

2
Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860-1674

Bajtek

MAGAZYN KOMPUTEROWY

NR 2(60)'91 CENA 7600

KUPIĆ, NIE KUPIĆ?
CO JEST CO
W ATARI?
ELEMENTARZ
PC



Desk Link

RSX
DLA KAŻDEGO
WATCH DOG
PACKET RADIO

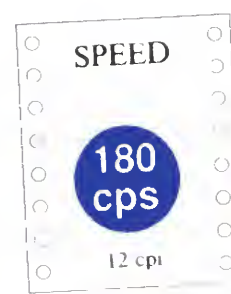
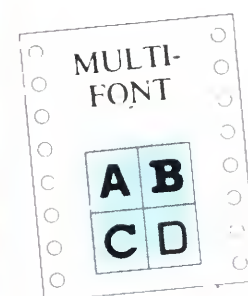
DRUKARKA STAR LC-20

— to nowa, szybsza LC-10



- Prędkość druku: 180 zn./sek.
- Jakość druku: standard oraz NLQ
- Traktor pchający
- „Parkowanie” papieru
- Automatyka oddzierania papieru
- Interfejs Centronics

Cena 2.500.000 (orientacyjna cena detaliczna)



DRUKARKA LC-200

— Star znów ustanawia nowy standard!

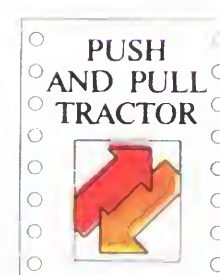
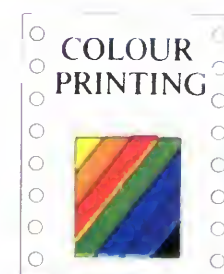
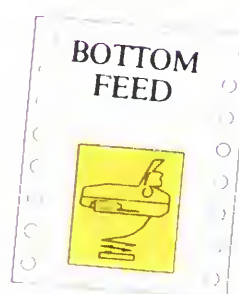
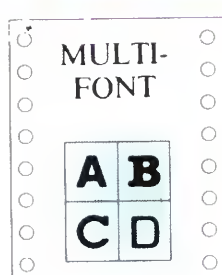
- Max. prędkość druku: 225 zn./sek.
- Druk kolorowy
- Możliwość podawania papieru od dołu
- Traktor pchający i ciągnący
- „Parkowanie” papieru
- Automatyka oddzierania papieru
- Interfejs Centronics

Cena 3.900.000

(orientacyjna cena detaliczna)



star
Twoja drukarka



ABC
D A T A
W A R S Z A W A

Przedstawicielstwo w Polsce
ABC Data Warszawa
ul. Waliców 13

tel. 24-11-43
24-78-35
tx. 816-423

Jadłospis

KOMPUTER ZABÓJCA . . . 4 MICRO MAGAZYN . . . 5 PO DZWONKU . . . 7

:Testy
:Kupić, nie kupić
:Jak to robią inni

KLAN ATARI . . . 10

:Co jest co w Atari
:Bzzzzzz
:Permutacje
:Errata
:Dwumian Newtona

DRUKOWANIE PRZY UŻY- CIU WŁASNEGO GENERATORA ZNAKÓW . . . 14

Klan Commodore . . . 16

:Szperacz dyskowy

KLAN SPECTRUM . . . 20

:Scrolling
:Smok i gęś
:Grafika trójwymiarowa —TP 3.0
:Język maszynowy cz.5

CO JEST GRANE . . . 22

:Dizzy
:Złota super trzydziestka
:SOS
:Serve & Volley
:P-47

MONITOR BIAZET MMK 127 —

TEST . . . 26

KLAN AMSTRAD . . . 27

:RSX dla każdego
:CP/M Plus — procedury systemowe w TP
:Procedury systemowe Amstrada cz. III

KLAN IBM . . . 30

:Elementarz PC
:Parking strzeżony
:Poprawka z grafiki
:Kompilatory Fortranu
:Desk-Link — oprogramowanie sieciowe
:Polskie litery

MAM DRUKARKĘ I . . . 35

ZRÓB TO SAM . . . 36

:Układ Watch-Dog
:Miernik pojemności

WSZYSTKO DLA WSZYSTKICH . . . 37

WESTER W POLSKIM STYLU . . . 38

DROBI BAJTKU . . . 41

PACKET RADIO . . . 44

W strukturze cen sprzętu komputerowego nastąpiły w zeszłym roku korzystne zmiany. Jeszcze kilka lat temu najtańszy 8-bitowy komputer stanowił równowartość kilkunastu albo kilkudziesięciu średnich wynagrodzeń miesięcznych. Obecnie nawet dość rozbudowany zestaw typu IBM PC jest kwestią tylko kilku pensji.

Obserwując rynek komputerowy roku 1990 można było zauważyć bogatą ofertę sprzętu profesjonalnego dostępnego dla instytucji i firm. Jednocześnie zaskakiwał zupełny brak podobnej oferty dla użytkowników indywidualnych. Łatwiej było nabyć zestaw AT 386 z dyskiem twardym 200 MB i kartą Super VGA niż jakiegokolwiek komputer domowy.

Większość odbiorców indywidualnych kupowała komputery typu IBM PC lub Commodore Amiga. Sprzęt 8-bitowy, nabywany w niewielkich ilościach głównie z „drugiej ręki”, znajdował zastosowanie u osób, które z powodów finansowych nie mogły nabyć komputera ze stacją dysków. Ze względu na ceny w sklepach Pexxu, CSH i innych placówkach handlowych, doszło do sytuacji, w której zakup np. komputera Atari ze stacją dysków (ok. 350 USD) odpowiadał cenowo najtańszej konfiguracji typu IBM PC (ok. 400 USD za zestaw z monitorem 14", stacją dysków 5.25" i z pamięcią RAM 640 KB).

Należy sądzić, że w obecnym roku tendencje te ulegną wzmocnieniu. Instytuty naukowe, biura projektowe, administracja i urzędy państwowe będą kupować coraz szybszy sprzęt klasy AT 386, 486 lub nawet stacje typu workstation, przy dużym nacisku na sieci i nowe silniejsze systemy operacyjne. W przypadku rynku odbiorcy prywatnego możemy oczekiwać poszerzenia oferty o komputery domowe klasy IBM PC (np. Sinclair PC 200 firmy Amstrad), a także większą powszechność modelu Commodore Amiga i Atari ST. W niektórych pracach specjalistycznych — mała poligrafia (DTP) i projektowanie wspomagane komputerowo (CAD) — mogą znaleźć zastosowanie niewielkie ilości sprzętu firmy Apple. Z kolei osoby pracujące w wolnych zawodach: literaci, dziennikarze, adwokaci będą prawdopodobnie zainteresowani komputerami klasy laptop lub jeszcze mniejszymi typu „książkowego” (ang. note-

book). Sprzęt 8-bitowy (małe Atari, Spectrum, Amstrad) będzie prawdopodobnie już tylko domeną rynku komputerów używanych.

Opisane tendencje wymuszają zastanowienie się nad tematyką, która ma dominować w tego-rocznych numerach Bajtka. Na pewno potrzebne będą podstawowe materiały dotyczące sprzętu 8-bitowego; duża część tych komputerów trafi w ręce osób, które nigdy nie miały z nimi do czynienia, a będą chciały wiedzieć co warto kupić, a czego nie. Bardziej zaawansowani użytkownicy tej klasy sprzętu będą zainteresowani opisem ciekawych programów użytkowych (edytory, arkusze kalkulacyjne, bazy danych) i ewentualnych rozszerzeń hardware'owych.

Ze względu na rosnącą rolę standardu IBM PC potrzebne będą informacje o istniejących emulatorach systemu MSDOS na 16-bitowe komputery Commodore Amiga i Atari ST. Z kolei wszystkim użytkownikom sprzętu komputerowego wyposażonego w stacje dysków przydadzą się informacje o programach umożliwiających wymianę tekstów i innych zbiorów między różnymi komputerami — tymi w pracy i w domu. W tym miejscu należy zwrócić uwagę na rosnącą rolę formatu 3.5" — coraz więcej komputerów jest wyposażonych w takie napędy.

A co w tym numerze? W Klanie Atari i IBM trochę materiałów podstawowych: „Co jest co w Atari” i „Elementarz PC”. Klan Commodore nie osiągnął jeszcze docelowej objętości 3–4 stron, ale prezentowany w nim materiał — „Szperacz dyskowy” — powinien zadowolić nawet najwybredniejszych. W Klanie Amstrad artykuł o RSX-ach w systemie AMSDOS jest rekompensatą dla osób, których ciekawość pobudziły prezentowane wcześniej informacje o podobnych rozszerzeniach dla CP/M-u.

Z tekstów ogólnych chcielibyśmy zwrócić uwagę Czytelników na materiał pt. „Packet Radio”, poświęcony nowej modzie, jaką jest krótkofalarstwo komputerowe. Temu zagadnieniu i podobnym, związanym z wymianą informacji między komputerami chcemy poświęcić więcej miejsca w następnych numerach naszego pisma.

Jarosław Młodzki

PRENUMERATA

Przyjmujemy wpłaty na prenumeratę wszystkich wydawanych przez nas pism. Prenumerata rozsyłana jest pocztą. Wpłaty prosimy dokonywać na nasze konto:

Spółdzielnia „Bajtek”
Bank „Agrobank S.A.”
479994-1834-131
ul. Grochowska 262
04-398 Warszawa

Po dokonaniu wpłaty należy wysłać na adres redakcji kopię przekazu oraz dokładne dane prenumeratora: pełny

adres, informację o tytułach na które została dokonana wpłata (oraz ewentualnie numer prenumery).

Pierwsze możliwe numery czasopism wyślemy w ciągu 6–8 tygodni od otrzymania informacji. Oferta jest ważna do 31.03.91.

Ceny prenumerat:

tytuł	3	6	12
Bajtek	X	45600	91200
Moje Atari	18000	36000	X
Top Secret	21600	43200	x

Bajtek

MAGAZYN KOMPUTEROWY

Redakcja:
ul. Wspólna 61,
00-687 Warszawa,
tel. 21-12-05

Redagują:
Redaktor naczelny — Jarosław Młodzki
Sekretarz redakcji — Marcin Przasnyski
Opracowanie graficzne — Wanda Roszkowska
Zdjęcia — Leopold Dzikowski
Szefowie klanów:
Amstrad — Jonasz Mayer
Atari — Wojciech Zientara
Commodore — Klaudiusz Dybowski
IBM — Marcin Borkowski
Spectrum — Maciej Pietras
Co jest grane — Łukasz Czekajewski
Edukacja — Tadeusz B. Mańk
Micro Magazyn — Janusz Jarmoch
Stali współpracownicy:
Grzegorz Bujanowski
Jarosław Burczyński
Marek Czarkowski
Piotr Kos
Robert Magdziak
Waldemar Nowak
Andrzej Pilaszek
Mieczysław Płacheta
Maria Radzimińska
Marek Sawicki
Piotr Sumara
Michał Szokoło
Stanisław Winiecki

Wydawca:
Spółdzielnia „Bajtek”
ul. Wspólna 61
00-687 Warszawa

Skład i druk:
Prasowe Zakłady Graficzne w Ciechanowie
Fotoskład: Grażyna Kurzątkowska
Montaż: Grażyna Ostaszewska
Korekta: Maria Krajewska
Teresa Rutkowska
Nakład: 107 tys. egz. Zam. 84260.

Celem ułatwienia zainteresowanym kontaktów z załogami poszczególnych klanów, stworzyliśmy system dyżurów szefów klanów. Najszybsze będzie dzwonicie w podanych godzinach w następujące dni:
Amiga — piątek 15.00 — 17.00
Amstrad — piątek 16.00 — 18.00 tel. 284594
Atari (Moje Atari) — wtorek, środa, czwartek 10.00 — 14.00
Co jest grane (Top Secret) — środa 12.00 — 16.00
Commodore — środa, 15.00 — 18.00
IBM — czwartek 15.00 — 18.00
Po dzwonku — wtorek 12.00 — 16.00
Spectrum — wtorek 14.30 — 16.00



PROGRAM MOŻE ZABIĆ

rozmowa
z prof. Andrzejem Blikle,
Prezesem
Polskiego
Towarzystwa
Informatycznego

Większość katastrof, jakie zdarzyły się w amerykańskim programie kosmicznym, została spowodowana błędami oprogramowania

— **Czy programista może ubiegać się w Polsce o uzyskanie znaku jakości dla swojego programu?**

Niestety, nie. W niektórych krajach zachodnich działają już jednak instytucje dokonujące homologacji i oceniające jakość programów komputerowych. Producenci nie mają obowiązku zgłaszania im do oceny swych produktów, ale mogą ubiegać się o „znak jakości”, który zwiększa szanse rynkowe. Ocena obejmuje nie tylko jakość techniczną programu, ale i odpowiedź na pytanie, czy program nie narusza cudzych praw autorskich.

Inne sposoby to atestowanie producenta i nadawanie programistom stopni zawodowych. Zajmują się tym często organizacje o charakterze stowarzyszeniowym.

— **A jak programista może zagwarantować sobie, że jego program będzie poprawnie funkcjonował?**

Od roku 1968, kiedy Komitet Naukowy NATO zorganizował w Europie konferencję na temat „inżynierii oprogramowania”, prowadzone są na bardzo szeroką skalę badania naukowe zmierzające do określenia reguł, według jakich powinien postępować programista tworząc program.

Inżynier konstruktor wie, jakimi zasadami powinien się kierować projektując np. most. Projekt mostu to rysunek techniczny wraz z komentarzem, jakich użyć materiałów, w których miejscach wystąpią największe obciążenia itp. Jeszcze przed postawieniem budowy sprawdza się, używając do tego matematyki, czy projekt jest poprawny — czy most się nie zawali.

W informatyce nie robi się rysunków i obliczeń, tylko pisze dwadzieścia, dwieście, dwadzieścia tysięcy stron, że ma być most, jak ma wyglą-

mie kosmicznym, została spowodowana błędami oprogramowania.

— **Jak w takim razie możemy wierzyć programistom? Popelniają błędy, nie znają zasad, jakimi powinni się kierować...**

Tak źle też nie jest. Programiści znają wiele zasad poprawnego budowania programów, zasady te jednak nie opierają się na matematyce, a więc nie dają gwarancji poprawności, a jedynie pewną jej szansę. Nie jest ona wystarczająco duża, musimy się jednak z tym faktem na razie pogodzić. Nie sposób przecież wyobrazić sobie dzisiejszej cywilizacji bez udziału informatyki.

Tylko w roku 1989 na rynku Europejskiej Wspólnoty Gospodarczej sprzedano sprzęt i oprogramowanie za łączną sumę 133 miliardów dolarów. Rynek w zasadzie eliminuje brakorobów. Mimo to zdarzają się dramatyczne błędy. W RFN program komputerowy zabił pacjenta aplikując mu zbyt dużą dawkę promieniowania.

— **Czy zatem istnieją już metody matematyczne pozwalające na ocenę jakości programu PRZED oddaniem go w ręce użytkownika?**

Poszukiwaniem takich metod zajmuje się informatyka teoretyczna. Mimo że zrobiono już wiele w tym kierunku, nie wypracowano jeszcze takich metod, jakimi posługują się specjaliści np. od budownictwa.

Nadal najczęstszym sposobem sprawdzenia jakości programu i usunięcia błędów jest uruchamianie go na różnych poziomach. Najpierw uruchamiamy i testujemy pojedyncze procedury, potem kompletne bloki funkcyjne, wreszcie cały system. Niestety jest to metoda bardzo kosztowna, a przy tym koszt znalezienia i usunięcia błędu rośnie z etapu na etap średnio dziesięciokrotnie. Usuwanie błędów u użytkownika kosztuje jeszcze więcej.

— **Jednak zgodnie ze słynnym aforyzmem Dijkstry — przy pomocy testowania można jedynie udowodnić obecność błędu, nie można natomiast takiej obecności wykluczyć.**

... Nic nie daje gwarancji absolutnych. Jednak wyniki jakie dają obliczenia i testy w „normalnej” inżynierii są bez porównania wyższe niż w przypadku programów. Prototypy samolotów na ogół nie spadają na ziemię... a programy na początku „padają” a potem „kuleją”. Ten problem narasta wraz ze stopniem ich skomplikowania szczególnie w chwili, gdy wprowadza się techniki programowania współbieżnego, tzn. gdy kilka, kilkanaście, lub kilkaset programów współpracuje ze sobą. A jest to technologia, która zaczyna być masowo stosowana przy tworzeniu dużych systemów. W tych przypadkach testowanie nie gwarantuje prawie żadnej wiarygodności programów.

— **Czy zatem powinniśmy nauczyć się żyć z błędami w programach komputerowych licząc na Opatrzność?**

Na razie tak. Z drugiej strony trzeba jednak przyznać, że to wszystko jakoś się kręci. A ponadto, pracujemy nad naprawą tego stanu rzeczy. Sam zajmuję się matematycznymi podstawami inżynierii oprogramowania i jestem przekonany, że jesteśmy na dobrej drodze do opracowania przyzwoitych matematycznych metod dla tej inżynierii.

Marek Czarkowski

dać, gdzie ma stać, jaką ma mieć wytrzymałość, itp. Potem zjawia się ekipa, czyta opis mostu i przystępuje do budowy. Kiedy most jest gotowy, przychodzi próba obciążenia (uruchomienia programu) i wtedy most — tzn. program — z reguły się zawala.

Praktycznie nie zdarza się, aby bezpośrednio po napisaniu programu działał poprawnie. Reguła ta może nie dotyczyć jedynie bardzo małych programów.

Kiedy tworzy się duży system software'owy, to około 60 procent budżetu producenta wydawanych jest na testowanie i usuwanie błędów. Mimo to błędy w programach pozostają. Czy wie Pan, że większość katastrof, jakie zdarzyły się w amerykańskim progra-

KOMPUTEROWE AKCESORIA

Jeszcze parę lat temu komputer można było kupić tylko na giełdzie. Najbliższy, dobrze zaopatrzony sklep znajdował się w Berlinie Zachodnim. Teraz z kupnem komputera nie ma już tak dużych problemów. Różne firmy oferują na tyle bogaty asortyment sprzętu, że każdy wybierze dla siebie coś odpowiedniego. Znacznie gorzej wygląda sprawa akcesoriów, które dla użytkownika Amigi, Atari, czy PC AT są tak niezbędne, jak dla kierowcy benzyna i płyn do mycia szyb. Nasi handlowcy muszą przemyśleć ten problem, aby klienci nie musieli wyjeżdżać do RFN po różne drobiazgi.

Kaufhof, oraz wiele innych, dużych sieci domów towarowych w RFN, traktują ten temat znacznie poważniej. Klient, który poszukuje komputera, nie odejdzie z kwitkiem. W wyspecjalizowanych stoiskach można natychmiast zrealizować każde marzenie. Przy okazji zaopatrzymy się w drukarkę, dyskietki, taśmy barwiące, fachowe książki i programy.

Nie brak również innych, pożytecznych akcesoriów, chociażby pojemników na dyskietki. Zgrabny, estetyczny system pudełek jest w stanie pomieścić nawet największy zbiór oprogramowania i danych. W każdej skrzyneczce mieści się 80 dyskietek 3,5". W zamkniętych na kluczyk szufladkach znajdują się kolorowe przekładki, które ułatwiają utrzymanie porządku. Pudełka można ze sobą łączyć, a w razie potrzeby zbudować nawet spory regał.

(J)



PRZED STARTEM
DO REGAT

Bez udziału komputerów nie jest możliwe zwycięstwo w żadnych liczących się w świecie regatach. Sukces żeglarzy uzależniony jest nie tylko od ich umiejętności. W poważnym stopniu decyduje o nim najnowocześniejsza technika obliczeniowa.

Walka o metry i sekundy na mecie zaczyna się już w biurach konstrukcyjnych. Potężne komputery uzbrojone w software najnowszej generacji optymalizują każdy element regatowego jachtu. Muszą one wyliczyć najkorzystniejszy kształt kadłuba, sprawdzić wytrzymałość masztów, zaprojektować sprawne aerodynamicznie żagle. Potem taki komputerowy model jest poddawany setkom godzin prób na programach symulacyjnych, które pozwalają przewidywać jego zachowanie w rzeczywistych warunkach. Po zakończeniu etapu obliczeń idą w ruch plotery, które kreślą rysunki nowego jachtu.

W stoczni stosowane są materiały, którymi nie pogardziłaby nawet technika kosmiczna. Skomplikowane technologie molekularne gwarantują wysoką wytrzymałość tworzyw sztucznych, które są używane do budowy kadłuba. Nawet szcycie żagli nie odbywa się ręcznie. Poszczególne elementy trójwymiarowej czaszy żagla, tzw. banty, muszą być bardzo precyzyjnie skrojone. Materiał cięty jest na specjalnych, sterowanych komputerem ploterach, które zamiast pisaka wyposażone są w nóż laserowy.

I wreszcie start do wyścigu. Wnętrze jachtu nasyconego jest elektroniką, nie mniej niż kokpit nowoczesnego samolotu. Ponad dwieście urządzeń pomiarowych stale kontroluje położenie steru, takielunek, ustawienie żagli, przemieszczenie balastu, przechył, siłę i kierunek wiatru. Komputer bez przerwy przelicza te dane i podaje optymalne korekty. Prowadzenie nawigacji ułatwia cały system skomputeryzowanych urządzeń: radary, echosondy, logi, wiatromierze.

Jedno z podręcznych urządzeń, które wspomaga sternika, prezentujemy na zdjęciu. Potrafi ono natychmiast podać długość i szerokość geograficzną, kurs, prędkość jachtu i wiele innych przydatnych parametrów.

(J)

TYLKO MODULA

W roku 1987 w kalifornijskiej miejscowości Mountain View pięciu sfrustrowanych programistów założyło własny interes. Na czele zespołu stał **Niels Jensen** — pozostali to **George Barwood**, **Jesper Boelsmand**, **Nigel Hicks** i **Ole Poulsen**.

Znali się bardzo dobrze, wcześniej wszyscy pracowali w firmie **Borland**, a **Niels Jensen** był nawet jednym z jej założycieli i współautorem takich „dzieł” jak **Turbo Pascal**, **SideKick** i **SuperKey**.

Zdobyte wcześniej doświadczenie pozwoliło im na rozkręcenie interesu.

Bardzo szybko firma **Jensen & Partners International** stała się znana na rynku ze znakomitej jakości kompilatora języka **Modula** — **TopSpeed Modula** — 2.

To co proponują jest lepsze od produktów **Borlanda** i **Microsoft** — napisał w kwietniowym numerze „**Computer Shopper**” z roku 1988 **How Colingburne**, a **Kent Porter** z „**Dr. Dobbs Journal**” dodał, że u niego „**TopSpeed** wywołuje owacje na stojąco”

Dlaczego produkty „Jensena i kumpli” otrzymują tak dobre oceny?

Użytkownikowi oferowane jest w pełni zintegrowane środowisko. Dobry edytor tekstu, **SmartLinker** wybierający z modułów tylko używane procedury, biblioteka z pełnym kodem źródłowym i możliwość korzystania z grafiki opracowanej przez firmę **JPI** lub z bibliotek **Borlanda**. Ciekawym uzupełnieniem jest **Visual Interactive Debugger** umożliwiający uruchamianie i śledzenie programów także współbieżnych.

Dzięki tym zaletom **TopSpeed Modula** — 2 bardzo szybko — jak podkreślają recenzenci — stał się standardem przemysłowym.

Kolejnym przebojem **JPI** okazał się **TopSpeed C** — kompilator modnego obecnie języka **C**. Latem tego roku zapowiedziano wprowadzenie na rynek kompilatorów **Pascala** i **C++** o nazwach **TopSpeed Pascal** i **TopSpeed C++** a na przyszły rok „Partnerzy” obiecują kompilator języka **Ada** — **TopSpeed Ada**.

Firma obiecuje pełną zgodność „swojego” **Pascala** ze standardem **Borlanda** i chyba można wierzyć ludziom, którzy współtworzyli sukces w/w firmy.

W firmach amerykańskich bardzo często zdarzają się odejścia ludzi bądź zespołów, którzy nie zgadzając się z polityką kierownictwa odchodzą i zakładają własny interes aby realizować pomysły, które często przynoszą ogromne pieniądze.

Tak było w przypadku **Intela**, **Atari**, **Apple** i setek innych przedsiębiorstw które swój początek biorą z twórczego niezadowolenia pracowników.

Ciekawe, czy kiedy ex-Borlandowcy opijają kolejny sukces ci co zostali zastanawiają się nad tym ile stracili.

Marek Czarkowski



NAJMNIJSZY ZENITH

Amerykańska firma Zenith Data Systems oferuje od sierpnia 1989 roku jeden z najmniejszych komputerów typu laptop o nazwie Zenith MinisPort. Urządzenie to waży nieco więcej niż 2.5 kg, ma rozmiary 318×249×34 (mm) i kosztuje około 2400 USD.

Jednostkę centralną stanowi procesor Intel 80C88 pracujący z częstotliwością 4.77 lub 8 MHz. W pamięci ROM o pojemności 360KB zainstalowano system operacyjny MS DOS 3.3 razem z programem FastLynx umożliwiającym wymianę plików z innymi komputerami. Pamięć RAM wynosi 1MB i jej część może być wykorzystywana jako ramdysk lub pamięć typu EMS. Możliwe jest jej dalsze rozszerzenie o następny megabajt.

Standardowym wyposażeniem jest złącze Centronics i 9-igłowe złącze RS 232C. Nietypowy jest rodzaj zastosowanej pamięci masowej — są to dwucalowe dyskiety o pojemności 720 KB. Dzięki nim było możliwe zmniejszenie wagi i rozmiarów opisywanego laptopa.

Opcjonalnie oferowany jest model 1200 bodów i dodatkowa stacja dysków (3.5" lub 5.25"). Ciekokrystaliczny ekran podświetlany od tyłu (backlit LCD) pracuje w trybie CGA. Możliwe jest podłączenie dodatkowego monitora w standardzie TTL lub analogowym. 12-woltowy akumulator pozwala na 3 godziny pracy urządzenia.

(JM)



DRUKARKOWE WARIACJE

Niezmiernie firma Star Micronics wciąż ulepsza produkowane przez siebie drukarki, wprowadzając na rynek coraz to nowe modele. Ostatnim innowacyjnym hitem jest drukarka LC-200, oferowana za ok. 3.9 mln zł.

Jest to konstrukcja całkowicie nowa i posiadająca możliwości spore, jak na produkt o niewielkiej cenie. Druk możliwy jest od górnej do dolnej krawędzi kartki; arkusz A4 można umieścić pionowo lub poziomo. Papier we wstędku może być wprowadzany standardowo z tyłu lub z dołu drukarki i kierowany na traktor o przełączalnym trybie pracy — pchający/ciągnący.

Ciekawostką jest możliwość zainstalowania taśmy barwnej, co pozwala na druk w 7 kolorach! Obie taśmy — czarna i kolorowa — dostarczane są fabrycznie. Nie będzie kłopotu z regeneracją tej pierwszej, ta druga więc będzie prawdopodobnie używana od święta.

Prędkość druku w trybie draft wynosi 225 zn/sek. bufor 16K znacznie poprawia komfort pracy i zbliża LC-200 do produktów klasy drukarek przemysłowych FR-10.

Podobnie jak LC-20, drukarka LC-200 posiada wbudowane polskie znaki zgodnie ze standardem Mazovii. (KM)



MicroEye

Do jednego z gniazd rozszerzających komputera PC można przyłączyć kartę MicroEye. Dzięki temu pożytecznemu urządzeniu wczytamy do pamięci dowolny, kolorowy lub jednobarwny, obraz. MicroEye współpracuje z magnetowidem, odtwarzaczem dysków laserowych lub kamerą wideo. W trybie barwnym mamy do dyspozycji teoretycznie aż 16.7 miliona kolorów. Dalsze wykorzystanie takiej ogromnej palety barw zależy już tylko od możliwości karty graficznej komputera.

Do obróbki obrazu sfotografowanego przy pomocy MicroEye musimy zastosować odpowiednie oprogramowanie. Każdy obrazek można nagrać na dysk, dołączyć do zbiorów opracowywanych w pakiecie DTP (np. Ventura publisher), no i oczywiście modyfikować i łączyć z tekstem w dowolny sposób.

(J)

Na podst. Personal Computer World

KOMPUTER W BIAŁYM DOMU

Cambridge Z88 jest komputerem idealnie nadającym się do pracy w klasach szkolnych, bibliotekach i podczas konferencji prasowych. Zawdzięcza to wyjątkowo „cichej” klawiaturze i niewielkim wymiarom.

Jerry Pournelle — dziennikarz amerykańskiego magazynu komputerowego „Byte” wizytując w roku ubiegłym Biały Dom zabrał ze sobą Z88 w celu sporządzenia notatek. Wywołało to sensację — Pournelle jako pierwszy zdecydował się na tego rodzaju „ekwipunek”.

„Ojcem” Z88 jest sławny sir **Clive Sinclair** — twórca rodziny komputerów Spectrum.

Cambridge Z88 to od 32 KB do 1.5 MB pamięci RAM i oprogramowanie: edytor tekstu, baza danych, kalkulator, notatnik, arkusz kalkulacyjny itp. Z88 wyposażony jest także w oprogramowanie pozwalające na komunikację, tak z komputerami klasy IBM PC, jak i Macintosh.

Polskim dziennikarzom na razie nie polecam tego modelu — edytor tekstu nie posiada (jeszcze) polskich znaków.

(MC)

TESTY TESTY TESTY

Kontrola wiadomości jest chyba najtrudniejszą częścią pracy nauczyciela.

Nie dość, że ciężko jest podjąć decyzję, czy odpowiedź już zasługuje na „dostateczny”, czy jeszcze nie, to na dodatek musimy liczyć się z możliwością posądzenia o niesprawiedliwą ocenę. Zresztą, jak wie każdy, ktc choć raz coś oceniał, trudno jest osiągnąć bezwzględną obiektywność. Sama ocena, może też zależeć od takich ulotnych warunków, jak np. samopoczucie nauczyciela. By uniknąć tych wszystkich kłopotów, wymyślono testy wielokrotnego wyboru. Dziś wiemy, że testy takie nie są najlepszą formą kontroli wiadomości, ale mimo to są z powodzeniem stosowane na wielu egzaminach.

Test wielokrotnego wyboru polega na tym, że uczeń otrzymuje treść pytania i kilka możliwych odpowiedzi, z których tylko jedna jest prawdziwa. Za każdą prawidłową odpowiedź przyznawane są punkty. Łączna suma zdobytych punktów określa poziom wiedzy i od tego poziomu zależy ocena.

Taka struktura testów powoduje, że można je łatwo przygotować w postaci programu komputerowego. Do przedstawienia pytań i propozycji odpowiedzi służy ekran, zaś udzielanie odpowiedzi polega na zwykłym wciskaniu pojedynczych klawiszy.

Testujące programy komputerowe mają wiele dodatkowych zalet. Komputer może losować kolejność pytań (uniemożliwia ściąganie!), wybierać tylko pytania z określonego działu czy o zadanym stopniu trudności.

Przedstawiony obok program jest prostym przykładem programu testującego. Brak wielu wymienionych opcji programowych jest spowodowany wyłącznie brakiem miejsca na łamach naszego pisma. Pozwala on jednak nauczycielowi na przeprowadzenie przygotowanego wcześniej sprawdzianu.

Program składa się z dwóch części: szyfrującej i testowej. Treść testu

przygotowywana jest przez nauczyciela wcześniej, jako plik tekstowy, za pomocą dowolnego edytora (np. tego samego edytora Turbo Pascala, na którym wpisano treść). Niestety, ponieważ w każdym programie najczęściej miejsca zajmują procedury sprawdzające poprawność, w naszym przykładzie zrezygnowaliśmy z nich. Powoduje to, że plik danych musi być poprawny i spełniać następujące warunki:

— rozszerzenie nazwy musi brzmieć 'TXT'

— pierwsza linia musi brzmieć 'TEST'

— pytania muszą być oddzielone jedną pustą linią

— pytania muszą mieć następującą formę:

— pierwsza linia: treść pytania

— druga, trzecia i czwarta linia:

treść odpowiedzi

— piąta linia: litera „A”, „B”, „C”,

oznaczająca prawidłową odpowiedź

— szósta linia: liczba punktów

przyznana za to pytanie.

Przykładowy tekst, składający się z trzech pytań dotyczących historii, przedstawiony jest na rys. 1.

Po napisaniu tekstu należy uruchomić program „SZYFR” — listing 1. Program zapyta o nazwę pliku szyfrowanego i hasło kodujące tekst. Następnie zostanie na dysku utworzony plik z rozszerzeniem 'TES', będący właściwymi danymi dla ucznia. Cały ten skomplikowany proces ma na celu tylko utrudnienie uczniom ściągania.

Gdy mamy gotowy tekst w postaci zaszyfrowanej, możemy przystąpić do testu. Uruchamiamy program „TEST” i podajemy nazwę pliku i hasło (oczywiście to samo, co przy szyfrowaniu), a następnie zapraszamy do klawiatury ucznia. Po skończonym teście należy sprawdzić, ile punktów nasz podopieczny uzyskał.

Program jest na tyle przejrzysty, że nie wymaga instrukcji obsługi, opisy zaś i komentarze w tekście programu pozwalają na jego rozbudowę o dodatkowe, potrzebne użytkownikowi opcje.

T.B.M.

```
TEST
W ktorym roku odbyła sie bitwa pod Grunwaldem?
1408
1410
1516
b
2

Jak sie nazywal ostatni krol Polski?
Stanislaw August Poniatowski
Stefan Batory
Jan Kazimierz
a
3

Jak sie nazywal pierwszy krol Polski?
Mieszko I
Boleslaw Chrobry
Popiel
b
3
```

Rys. 1

LISTING 1

```
(*****
(**      T E S T Y - szyfrowanie      **)
(** Turbo Pascal >=3.0, (c)TBM 1990 **)
(***)

uses crt;      (* WYKASOWAC LINIE DLA WERSJI 3.0 *)

type
  string8 = string[8];
  string255= string[255];
var
  dane, daneSzyfr : text;
  nazwa, klucz    : string8;

function StartPlikow(nazwa:string8) : boolean;
var
  OK : boolean;
begin
  if pos('.',nazwa)>0 then
    delete(nazwa,pos('.',nazwa),length(nazwa));
  assign(dane,      nazwa + '.TXT');
  assign(daneSzyfr,nazwa + '.TES');
  {$I-} reset(dane); {$I+} (*****
    If IOresult<>0      (*SPRAWDZENIE ISTNIENIA*)
    then begin         (*I POPRAWNOSCI DANYCH *)
      OK:=false; (*****
      writeln('BRAK PLIKU');
      end
    else begin
      OK :=true;
      rewrite(daneSzyfr);
      end;
  startPlikow := OK;
end;

procedure szyfrowanie(klucz : string8);
var LiteraKlucza : integer;
  znak : char;
  kody : array[1..8] of byte;
  i : integer;
begin
  for i:=1 to 8 do
    kody[i]:=ord(klucz[i]);
  literaKlucza:=0;
  while not eof(dane) do
    begin
      literaKlucza:=literaKlucza+1;
      read(dane,znak);
      znak:=chr(ord(znak)+kody[literaKlucza]);
      write(daneSzyfr,znak);
      if LiteraKlucza>7 then LiteraKlucza:=0;
    end;
  close(dane);
  close(daneSzyfr);
end;

begin
  clrscr;
  write('NAZWA SZYFROWANEGO PLIKU (max 8 liter)');
  readln(nazwa);
  if startPlikow(nazwa) then
    begin
      write('PODAJ KLUCZ SZYFRU ');
      readln(klucz);
      while length(klucz)<8 do klucz:=klucz+' ';
      szyfrowanie(klucz);
    end;
end.
```

LISTING 2

```
(*****
(**      T E S T Y - odpytka      **)
(** Turbo Pascal >=3.0.(c) TBM 1990 **)
(***)

uses crt;      (* WYKASOWAC LINIE DLA WERSJI 3.0 *)
type
  string8=string[8];
  string255=string[255];
var
  znak:char;
  dane:file of char;
  klucz:string8;      (* HASLO KODUJACE DANE *)
  nazwa:string8;      (* NAZWA PLIKU DANYCH *)
  kody :array[1..8] of byte;(* INNA POSTAC HASLA *)
  pozKlucza:byte;
  pkt,pktglb:integer; (* PKT. UZYSKANE I OGOLEM *)

function startPliku(nazwa,klucz :string8):boolean;
var OK:boolean;
  i :integer;      (* PROCEDURA SPRAWDZAJACA *)
begin      (* ISTNIENIE PLIKU DANYCH *)
  for i:=1 to 8 do
    kody[i]:=ord(klucz[i]);
  if pos('.',nazwa)=0
  then assign(dane,nazwa+'.TES')
  else assign(dane,nazwa);
  {$I-} reset(dane); {$I+}
```



```

if IoResult<> 0
then begin
    OK:=false;
    writeln('BRAK PLIKU O TEJ NAZWIE');
end
else OK:=true;
startPliku:=OK;
pozKlucza:=0;
pkt:=0: pktglb:=0;
end;

function deszyfr(znak:char):char: (* PROCEDURA *)
begin (* DEKODUJACA ZNAK TEKSTU *)
    pozKlucza:=pozKlucza+1;
    deszyfr:=chr(ord(znak)-kody[pozKlucza]);
    if pozKlucza>7 then pozKlucza:=0;
end;

function KolejnaLinia:string255;
label koniecLinii; (* DEKODOWANIE KOLEJNEJ *)
var znak:char; (* PELNEJ LINII TEKSTU. *)
linia:string255; (* UZYTE ETYKIETY, CO NIE *)
begin (* JEST ELEGANCKIE. ALE *)
    linia:=''; (* BARDZO POPRAWIA *)
    if not eof(dane) then(* CZYTELNOSC *)
    begin
        koniecLinii: read(dane,znak);
        znak:=deszyfr(znak);
        if znak<>'#13'
        then begin
            linia:=linia+znak;
            goto koniecLinii;
        end
        else begin
            read(dane,znak);
            znak:=deszyfr(znak);
        end;
    end;
    KolejnaLinia:=linia;
end;

procedure piszPytanie;
begin
    writeln(KolejnaLinia);
    writeln;writeln;
    writeln('A: ',KolejnaLinia);writeln;
    writeln('B: ',KolejnaLinia);writeln;
    writeln('C: ',KolejnaLinia);
end;

procedure kolejnePytanie;
var znak:char;
    znak1:char;
    linia:string[255];
    punkty,kod:integer;
begin
    piszPytanie;
    linia:=kolejnaLinia;
    znak:=upCase(Linia[1]);
    repeat
        znak1:=readkey; (* read(kbd,znak1); dla v.3.0 *)
        znak1:=upCase(znak1);
        until znak1 in ['A','B','C'];
        val(kolejnaLinia,punkty,kod);
        if kod<>0 then punkty:=0;
        if znak=znak1
        then begin
            pkt:=pkt+punkty;
            sound(400); delay(70);
            gotoxy(1,18):writeln('DOBRA ODPOWIEDZ!');
            nosound;
        end
        else begin
            sound(800); delay(70);
            gotoxy(1,18);
            writeln('ZLE!Dobra odpowiedz to: ',znak);
            nosound;
        end;
        writeln:writeln('NACISNIJ DOWOLNY KLAWISZ');
        repeat until keypressed;
        pktglb:=pktglb+punkty;
    end;

procedure podajDane;
begin
    clrscr;
    write('Podaj nazwe pliku danych ');readln(nazwa);
    write('Podaj haslo '); readln(klucz);
    while length(klucz)<8 do klucz:=klucz+' ';
    clrscr;
end;

begin
    clrscr;
    podajDane;
    if startPliku(nazwa, klucz) then
        if kolejnaLinia<>'TEST' (*PIERWSZA LINIA*)
        then writeln('ZLE HASLO')
        else begin while (not eof(dane)) do
            begin
                kolejnePytanie;
                write(kolejnaLinia);
                clrscr;
            end;
        end;
        clrscr;
        gotoxy(1,10);
        writeln('Uzyskano ',pkt,' punktow na
            pktglb,' mozliwych');
    end;
end;

```

KUPIĆ, NIE KUPIĆ?

W serii artykułów pod wspólnym tytułem „KUPIĆ, NIE KUPIĆ?” chcemy przedstawić Państwu opisy popularnych w Polsce komputerów pod kątem ich przydatności w procesie edukacji. Zwięzły opis danych technicznych i możliwości ma na celu pomoc przy podejmowaniu decyzji dotyczącej wyboru komputera dla szkoły.

ZX SPECTRUM

1. Dane techniczne.

Komputer ZX Spectrum zrobił w Polsce oszałamiającą karierę, a to ze względu na niską cenę i dość dobre możliwości jak na tak prostą konstrukcję. Podstawowa konfiguracja posiada 48 KB pamięci RAM i 16 KB pamięci ROM z wbudowanym interpreterem BASIC-a. Spectrum posiada procesor Z-80A taktowany zegarem 3,5 MHz.

2. Konfiguracja.

Obraz uzyskuje się przez podłączenie Spectrum do zwykłego telewizora. Może być też użyty monitor, lecz tylko przez specjalny interfejs (schemat był publikowany m.in. w BAJTKU nr 9—10/90).

Jako pamięć zewnętrzną producenci przewidzieli magnetofon

kasetowy. Nie polecamy go jednak, gdyż jest on zawodny i powolny. Wiele dostępnych programów edukacyjnych jest mimo to rozprawdzanych na kasetach i trzeba używać ich w ten sposób.

Problem ten znika, gdy łączymy ZX Spectrum ze stacją dysków TIMEX. Obsługa poleceń stacji jest bardzo prosta, a specjalny system dyskowy TOS jest dość przyjazny dla użytkownika. Choć obecnie można mówić o zmierzchu komputerów ośmiobitowych, to ZX Spectrum właśnie dzięki stacji dysków przeżywa swą drugą młodość.

3. Możliwości wersji podstawowej.

Jak już wspominałem, Spectrum posiada wbudowany interpreter BASIC-a, który jest jednym

JAK TO ROBIĄ INNI?

rozmowa z prof. Helmutem Kühneltem z Instytutu Fizyki Teoretycznej Uniwersytetu Wiedeńskiego.

— W jaki sposób w Austrii wykorzystywane są komputery w procesie edukacji?

— Komputery wprowadzamy dla naszych uczniów w wieku ok. 15 lat, w liceach. Uczniowie ci, w większości przyszli studenci szkół wyższych mają obowiązkową naukę podstaw komputeryzacji w wymiarze 2 godz. tygodniowo. Naszym głównym celem nie jest nauka programowania, lecz wyrobienie umiejętności używania komputerów jako narzędzia pracy. Chcemy wpoić uczniom rozumienie wielkiego społecznego znaczenia rewolucji informatycznej. To nasz podstawowy cel lecz oczywiście krótkie, proste programy mogą pokazać metody stosowane w informatyce, więc uczniowie muszą nie tylko je rozumieć, ale także pisać. Muszą to umieć wszyscy uczniowie liceum, zwłaszcza ci, którzy wybierają się

na studia. Programy te pisane są w różnych językach. Może to być BASIC, ale także PASCAL, LOGO, czy inny. Wszyscy nasi uczniowie muszą znać edytory tekstu, bazy danych, arkusze elektroniczne. Muszą sobie zdawać sprawę, jak ważna jest znajomość tych programów w dalszym życiu i pracy zawodowej. Uczniowie zwiedzają zakłady przemysłowe, biura obsługi ludności i zapoznają się ze sposobami organizacji, przechowywania i wyszukiwania informacji.

Oczywiście mamy też klasy o specjalności informatycznej. Wybór takiej klasy jest całkowicie dobrowolny, a ich program jest nastawiony na naukę programowania. Po skończeniu takiej specjalności można ubiegać się o przystąpienie do egzaminu państwowego z programowania.

— Czy w swojej pracy nauczyciele posługują się specjalnymi programami edukacyjnymi?

— Na lekcjach informatyki nie. Komputer wykorzystywany jest do pokazania, jak można go wykorzystać w życiu codziennym, w pracy zawodowej. Dlatego też



Niestety, tylko Spectrum...

z lepszych interpreterów wśród komputerów tej klasy. Jedyną jego poważną wadą jest liniowy, nie ekranowy edytor. Jest to pewne utrudnienie dla programistów.

Jak na produkt myśli technicznej z 1982 r. Spectrum posiada zadziwiająco dobrą, kolorową grafikę wysokiej rozdzielczości. Ekran o parametrach 256 na 192 pkt. może wyświetlić 16 kolorów (osiem podstawowych i ich rozjaśnienie). Tryb znakowy nie jest rewelacyjny, bo oferuje tylko 22 wiersze po 32 znaki w każdym, niemniej znaki są wyraźne i czytelne. Istnieją edytory oferujące 64 znaki w wierszu, jednak pogarsza się w ten sposób czytelność tekstów.

4. Polskie litery.

Konieczność istnienia polskich

liter na komputerze w polskiej szkole nie podlega dyskusji. Na Spectrum jest dość łatwo zdefiniować litery specyficzne dla polskiego alfabetu (ą, ć, ę, itd.). Definicję taką możemy przeprowadzić dwójako: przez grafikę użytkownika (UDG), albo zmieniając cały zestaw znaków. Operacji tej może dokonać nawet niedoświadczony programista posługując się tylko instrukcją obsługi. Tak otrzymane litery są dostępne zarówno na ekranie, jak i w wydrukach.

5. Rozbudowa.

Komputer Spectrum został tak pomyślany, aby w podstawowej konfiguracji zawierał minimum urządzeń niezbędnych do jego prawidłowego funkcjonowania. Furtką zostawioną dla chcących zwiększyć jego możliwości jest

szyna krawędziowa, do której doprowadzono wszystkie sygnały procesora. Do złącza tego można (poprzez interface-y) dołączać różne urządzenia zewnętrzne, drukarkę, wszelkiego rodzaju urządzenia pomiarowe, sprzęgi MIDI, moduły muzyczne, syntezy mowy, mysz itd. Schematy interace-ów do tych urządzeń były i będą drukowane w Bajtku, a wiele z nich można po prostu kupić.

Kolejnym problemem zajmującym nas ze względu na przydatność w szkołach jest sieć komputerowa. Aby wykorzystać Spectrum do stworzenia lokalnej sieci komputerowej potrzebny jest produkt o nazwie Interface 1. Służy on do łączenia komputera z — niepopularną u nas — pamięcią masową Microdrive, a jednocześnie umożliwia połączenie od 2 do 64 Spectrum-ów w sieć. Sieci takie mogą być używane w szkole ze względu na swą uniwersalność, i na to, że istnieje możliwość sterowania pracą całej sieci za pomocą jednej maszyny.

6. Możliwości zakupu.

Gdzie można nabyć Spectrum i za ile? Obecnie, ze względu na wiek omawianego sprzętu nie jest prosto go kupić, mimo tego, że cena w stosunku do możliwości jest niska. Waha się ona (w zalecanej konfiguracji: komputer + stacja dysków + monitor) w granicach 220 dolarów. Kupić najlepiej na giełdzie (niestety bez rachunku).

7. Oprogramowanie.

Należy wiedzieć, że ZX Spectrum zawdzięcza swą popularność olbrzymiej ilości oprogramowania. Istnieje wiele programów

użytkowych (bazy danych, arkusze kalkulacyjne, edytory tekstów) i kompilatorów języków oprogramowania. Wśród nich wart jest podkreślenia m.in. język Turbo Pascal 3.0, dostępny po podłączeniu stacji dysków FDD 3000.

Stacja dysków FDD 3000 otwiera nam dodatkowo możliwość korzystania z olbrzymiej biblioteki programów pisanych dla systemu CP/M 2.2, w który to system operacyjny jest wyposażona.

Jeśli chodzi o programy edukacyjne, Spectrum jest komputerem, dla którego powstało ich chyba najwięcej. Większość polskich firm software'owych tworzących programy edukacyjne ma ten komputer na swojej liście. Wśród nich można znaleźć nie tylko wspomniane już programy użytkowe, których nauczanie zalecają wszyscy znani dydaktycy zajmujący się komputerami, ale wiele innych, tworzonych z nastawieniem na wykorzystanie w szkole. Szeroką gamę takich programów (głównie z chemii, fizyki, informatyki i matematyki) oferują np. firmy KOMPRED i VULCAN. Są to programy edukacyjne i kształcące, oraz testujące i pomocnicze. W szkole znajdą również z pewnością zastosowanie specjalistyczne programy dla administracji szkolnej. Programy te będziemy prezentować w przyszłości. Polecamy też planowany cykl artykułów poświęconych możliwości wykonywania doświadczeń szkolnych, w których Spectrum służy do zbierania i prezentacji danych.

Maciej Pietraś

wykorzystujemy ogólne, komercyjne programy.

— A jak są wykorzystywane komputery na innych lekcjach?

— Tu również staramy się używać programów komercyjnych. Nasi uczniowie w wieku 14 lat przechodzą tygodniowy intensywny kurs informatyki. Na kursie tym uczymy edytorów tekstu i innych komercyjnych programów. Są one później pomocne w nauce. I tak edytory tekstów są wykorzystywane na lekcjach języków. Germaniści i humaniści uczą się edycji i pisanie tekstów, korzystania z tekstów pisanych przez innych. W nauce języków obcych często korzystamy z gier typu „ADVENTURE”. Natomiast bazy danych są intensywnie eksploatowane przez historyków. Uczniowie używając tych baz poznają historię, zagadnienia społeczne, socjalne i in.

Innymi programami używanymi w szkole są programy pisane przez nauczycieli. Te krótkie i łatwe programy edukacyjne są często wykorzystywane na lekcjach, bądź w pracach własnych uczniów.

— Kto tworzy programy szkolne, i kto za nie płaci?

— Programy użytkowe są oczywiście tworzone przez firmy software'owe. Są to ogólnie znane

programy komercyjne.

Część finansów jest oferowana szkołom przez Ministerstwo. W kosztach partycypują również miasta subsydiujące szkoły, ale to szkoły same decydują, jak i na co wydać pieniądze.

Dużą wagę przywiązujemy do konkretnych aplikacji takich ogólnych programów, jak arkusze, czy bazy danych. Tworzą je nauczyciele pisząc „helpery” i pakiety pomocnicze czyniące program bardziej przyjazny dla użytkownika. Za te pakiety płaci Ministerstwo.

— Jak uczniowie przyjmują komputery?

— Bez entuzjazmu. Możemy wyróżnić dwie grupy: entuzjaści i inni. Entuzjaści są oczywiście zachwyceni komputerami w ogóle. Jednak chcą oni tylko grać, stosować nowe sztuczki programistyczne (wirusy są dla nas prawdziwym problemem). W żaden sposób nie potrzebują oni szkoły do nauki i nauczyciele im raczej przeszkadzają. Inni uczniowie traktują komputer jak jeszcze jeden środek dydaktyczny, taki jak rzutnik pisma, video itp.

— Jak wielu uczniów ma komputery i jakie?

— Oceniamy, że ok. 80—90%. Głównie są to COMMODORE 64. Ale mamy duże wahania: np. są

okręgi, w których liczba ta nie przekracza 10%. Powstaje problem, jaki system operacyjny stosować: czy systemy komputerów 8-mio bitowych (COMMODORE, ATARI, których też mamy sporo) czy MS-DOS, w którym pracują komputery szkolne. Po wielu dyskusjach przyjęto, decyzją Ministerstwa, naukę systemu profesjonalnego MS-DOS.

— Jak wygląda przygotowanie nauczycieli do wykorzystywania komputerów?

— Jest to jeden z naszych największych problemów: jak wytłumaczyć nauczycielom, szczególnie tym ze starszej generacji, tę potrzebę. Mamy krótkie, trzytygodniowe intensywne kursy, oraz długie, wielomiesięczne przygotowanie organizowane przez Ośrodki Kształcenia Nauczycieli.

Lepsza sytuacja jest wśród młodszych kadr. Studenci kierunków nauczycielskich są właściwie kształceni w dwóch kierunkach: swej własnej specjalności i informatyce. Uczni są oni tych samych programów, których będą potem uczyli swych uczniów. Prócz tego uczą się obsługi systemów operacyjnych, programowania. Bardzo ważna jest również nauka aplikacji gotowych programów, o czym już mówiliśmy.

Ten sam program obejmują rów-

nież kursy dla nauczycieli, seminaria i inne formy kształcenia. Kładziemy również nacisk na sprawy tajności danych personalnych i handlowych.

— Czy w Austrii szkoły są połączone w sieć?

— Przede wszystkim mamy videotext. Jest to sieć, w której terminalem może być komputer. W sieci tej dostępne są informacje (jest to sieć ogólna) również edukacyjne. Mamy oczywiście telefony (szczęściarze — przyp. red.) i w związku z tym możliwość łączności przez modemy. Ale takie połączenia są wolne i dużo kosztują, więc wiele szkół rezygnuje z tej formy. Wiele szkół wykorzystuje edukacyjną sieć dla komputerów APPLE. Mają dzięki niej łączność z całym światem i dostęp do najnowszych osiągnięć w dydaktyce.

— Jaką Pańskim zdaniem, drogą powinna pójść komputeryzacja szkolnictwa w Polsce?

— Za mało znam tutejsze warunki, by móc się autorytatywnie wypowiadać. Jeśli mogę przedstawić swoje zdanie, to myślę, że należy się bardziej skoncentrować na wykorzystaniu komputerów niż na nauce programowania.

— Dziękuję za rozmowę.

T.B. Mańk

CO CO jest CO CO

W A T A R I ?

„Jakie są różnice między Atari 800XL i 65XE?” „Proszę o podanie danych technicznych Atari 65XE i 130XE.” „Czy magnetofon CA12 jest lepszy od XC12 i na czym polega różnica?” „Jaka stacja dysków jest najlepsza do 65XE?” „Czy 520ST może współpracować z magnetofonem i telewizorem?”

Takich i podobnych pytań otrzymuję bardzo dużo. Świadczą one nie tylko o nieznajomości komputerów, ale również — niestety — o unikaniu przez naszych czytelników „ogromnego” wysiłku umysłowego, jakim jest przeczytanie instrukcji obsługi komputera (u kolegi, gdy się nie ma własnego) albo przejrzenie kilku starych numerów „Bajtki” (jak myślicie, po co w każdym numerze styczniowym jest spis treści z poprzedniego roku?).

Niemal wszystkie informacje, o które prosicie w swoich listach, były już wcześniej podawane w „Bajtku” i jego wydaniach specjalnych „Tylko o Atari”, a następnie w „Moim Atari”. Ponieważ jednak listów takich wcale nie ubywa, postanowiłem odpowiedzieć na nie hurtowo.

DWIE RODZINY

Zasadniczy podział komputerów Atari jest związany z zastosowanymi w nich procesorami. Atari XL i XE (tzw. „małe”) mają ośmiobitowy procesor 6502, ST („duże”) zaś — szesnastobitowy procesor 68000. Są to zupełnie odmienne komputery — różnią się one tak jak „maluch” od Poloneza. Nie można więc urządzeń i programów przeznaczonych dla jednego z nich wykorzystywać z drugim. Istnieje wprowadzicie program powodujący, że ST udaje XE (fachowo mówi się „emuluje”), ale korzyść z tego mniejsza niż z zainstalowania w polonezie silnika od „malucha”. Dlatego też dalszy opis tych komputerów musi być oddzielny dla każdej rodziny.

MAŁE JEST PIĘKNE

To popularne hasło jest niezupełnie prawdziwe, ale dobrze sprawdza się w życiu. Po prostu, nie każdego stać na mercedesa — gdyby tak było, nikt nie kupowałby „maluchów”, a wystarczy spojrzeć na ulicę... Atari XL i XE są dobrymi komputerami ośmiobitowymi i nie zaprzeczy temu nawet najbardziej zaprzeczający „komodorowiec” czy „trumniarz” (o ile widział kiedyś Atari w działaniu). Przy odrobinie umiejętności i chęci (oraz mając więcej niż odrobinę pieniędzy) można z niego wycisnąć prawie tyle, co z IBM-a. Nie mówię tu oczywiście o grach — nic nie jest w stanie zastąpić wysokiej rozdzielczości, dużej liczby kolorów i znacznie większej pamięci. Jakie więc są te „małe” Atari?

Wszystkie odmiany XL i XE mają taki sam system operacyjny (niewielkie różnice są istotne tylko dla profesjonalistów). Dzięki temu przeznaczone dla jednego z nich programy **muszą** działać na innym, chyba że zabraknie dla nich pamięci. Właśnie pojemność pamięci jest podstawową cechą różniącą poszczególne modele. Dru-

ga różnica polega na nieco innych złączach (tych mniej ważnych). Identyczne są następujące parametry komputerów Atari 600XL, 800XL, 65XE, 130XE, 800XE i XE Game System:

- procesor 6502,
- zegar taktujący 1,77 MHz,
- system operacyjny w pamięci ROM 16 KB,
- interpreter Atari Basic (ROM 8 KB),
- 5 trybów tekstowych,
- 11 trybów graficznych,
- maksymalna rozdzielczość 320 x 192 punkty,
- wybór z palety 256 kolorów,
- 4 generatory dźwięku,
- wbudowany modulator telewizyjny PAL,
- klawiatura QWERTY — 57 klawiszy,
- 5 klawiszy funkcyjnych,
- złącze szeregowe dla urządzeń zewnętrznych,
- dwa gniazda joysticków,
- gniazdo modułu ROM (cartridge'a),
- gniazdo antenowe,
- gniazdo monitorowe.

Najważniejszą różnicą jest pojemność pamięci operacyjnej. Podstawowy jej rozmiar to 64 KB. Więcej (128 KB) ma tylko 130XE, mniej zaś (16 KB) tylko 600XL. Tak więc programy lub ich wersje napisane specjalnie dla 130XE nie będą działały lub będą działały nieprawidłowo na pozostałych komputerach. Znacznie gorzej jest w przypadku 600XL. Tylko nielicznym programom wystarcza tak mała pamięć, a praca ze stacją dysków jest zupełnie niemożliwa; albo w pamięci zmieści się program, albo DOS — dla obu naraz nie ma miejsca. Bardzo popularne rozszerzenia pamięci nie powodują tu zwykle żadnych zmian — zgodność zostaje zachowana i wszystkie programy, które dotychczas działały, muszą działać nadal. Jeżeli jest inaczej, należy natychmiast zgłosić reklamację do autora rozszerzenia.

Pozostałe różnice dotyczą wyglądu, co oczywiście nie ma znaczenia dla programów oraz wspomnianych już wcześniej złączy. Chodzi tu o wyprowadzenie głównej magistrali systemu, czyli tzw. szyny równoległej. Pełne złącze równoległe mają tylko komputery XL. W 65XE dostępna jest tylko część tej szyny, dołączona do gniazda modułu ROM. Jej uzupełnieniem jest w 130XE i w najnowszych egzemplarzach 65XE dodatkowe złącze (ENHANCED CARTRIDGE), zawierające sygnały, których brakuje w złączu modułu.

MAGNETOFONY

Podobnie przedstawia się sprawa magnetofonów — są one całkowicie jednakowe pod względem współpracy z komputerem. Dla użytkownika różnią się one jedynie wyglądem i trwałością. Dla nikogo nie jest tajemnicą, że Atari 1010 ma słabe klawisze, a w XC12 łatwo rozregulowuje się ustawienie głowicy. Są to jednak tylko różne rozwiązania mechaniczne — w środku wszystkie magnetofony działają jednakowo, co nie oznacza, że są zbudowane z jednakowych części. Ale to kłopot dla serwisu, a nie dla właściciela.

Dane techniczne magnetofonów Atari 1010, XC11, XC12, XCA12 oraz CA12 są następujące:

- zapis stereofoniczny,
- prawy kanał dla sygnału komputerowego,
- lewy kanał dla dźwięku towarzyszącego,
- transmisja szeregową,
- prędkość transmisji 600 bodów (bitów na sekundę),
- automatyczna regulacja poziomu zapisu,
- sterowanie silnika przez komputer.

STACJE DYSKÓW

Nieco bardziej różnią się od siebie stacje dysków stosowane do „małych” Atari. Jest ich wiele, więc zajmę się tylko tymi, które występują w Polsce w większych ilościach. Są to: Atari 1050, Atari XF551, LDW Super 2000, California Access 2001 oraz będąca jeszcze w opracowaniu California Access 2002. W miarę rozwoju techniki komputerowej zmieniał się sposób zapisu na dyskietkach, a dokładniej — liczba informacji możliwych do zapisania na jednej dyskietce. Obecnie istnieją cztery różne gęstości zapisu stosowane w stacjach dysków dla Atari. Nie wnikając w szczegóły powiem tylko, że są to gęstości: pojedyncza (90 KB na jednej stronie dyskietki), rozszerzona (130 KB), podwójna (180 KB) oraz tzw. poczwórna, która jest w rzeczywistości gęstością podwójną, gdyż umożliwia zapisanie 360 KB na dwóch stronach dyskietki (bez jej odwracania).

Poszczególne stacje różnią się właśnie możliwością zapisu i odczytu dyskietek w różnych gęstościach. Atari 1050 umożliwia korzystanie tylko z gęstości pojedynczej i rozszerzonej, LDW 2000 i CA 2001 rozpoznają gęstość pojedynczą, rozszerzoną i podwójną, XF551 i CA 2002 zaś posiadają dwie głowice, mogą więc pracować także w gęstości poczwórnej. Inne, istotne dla użytkownika parametry tych stacji są jednakowe:

- dyskietki o średnicy 5¼ cala,
- transmisja szeregową,
- prędkość transmisji 19200 bodów,
- dwa złącza szeregowe Atari.

Do powyższych danych konieczne jest dodanie pewnej uwagi. Podana prędkość transmisji jest prędkością standardową. Można ją zwiększyć programowo. Razem ze stacjami LDW 2000 i CA 2001 sprzedawany jest DOS XL, który zawiera między innymi program SYNCHROMESH, umożliwiający dwukrotne zwiększenie szybkości przesyłania danych — do 38000 bodów. Podobny efekt uzyskuje się podczas pracy stacji LDW ze SpartaDOS X. Przy współpracy ze stacjami XF551 i CA 2002 system ten zwiększa prędkość transmisji aż do 70000 bodów.

Proszę zwrócić uwagę, że pisałem wszędzie „stacja ma możliwość pracy w jakiejś gęstości”, a nie „stacja pracuje w jakiejś gęstości”. Rzeczywista gęstość dyskietki zależy bowiem nie tylko od stacji, ale także od zastosowanego DOS-u (dyskowego systemu operacyjnego). Jeśli DOS nie potrafi rozpoznać jakiejś gęstości, to nawet

stawianie na głowie nie zmusi stacji do pracy w takiej gęstości. Na przykład, bardzo popularny DOS 2.5 rozpoznaje tylko gęstość pojedynczą i rozszerzoną, więc używanie go ze stacjami LDW, CA i XF jest nieporozumieniem. Jeszcze gorzej jest w przypadku dwustronnych stacji XF551 i CA 2002. Wszystkie możliwości tych stacji potrafi wykorzystać jedynie SpartaDOS X. Zapis dwustronny jest ponadto możliwy przy użyciu MYDOS i DOS XE.

INNE URZĄDZENIA

Spośród pozostałych urządzeń peryferyjnych wykorzystywane są przez wszystkich monitory (telewizory), a przez nielicznych drukarki. Inne peryferia występują w Polsce w ilościach śladowych. Wbrew pozorom nawet podłączenie telewizora sprawia kłopoty wielu użytkownikom. Komputery Atari są bowiem przystosowane fabrycznie do współpracy z kolorowymi telewizorami w systemie PAL. Użycie telewizora w systemie SECAM nie pozwala na uzyskanie kolorów i dźwięku. Poza zakupem nowego telewizora istnieją jeszcze dwa inne rozwiązania tego problemu: przestrojenie modulatora w komputerze lub zainstalowanie dekodera PAL w telewizorze. Pierwszy sposób nie wymaga objaśnień. Natomiast w drugim przypadku często zdarzają się nieporozumienia. Systemy PAL i SECAM różnią się sposobem kodowania koloru i częstotliwością różnicową fonii. To techniczne określenie oznacza różnice pomiędzy częstotliwościami, na których nadawany jest obraz i dźwięk. Wielu „fachowców” — licząc na nieświadomość klienta — instaluje niepełny dekodery PAL. Zmienia on tylko sposób kodowania koloru, lecz nie zmienia częstotliwości różnicowej fonii. Komputer wykrywa to jednak bezbłędnie i za nic nie wydobędzie z siebie głosu na takim dekoderyze. Ponieważ o łączeniu komputera z telewizorem i monitorem pisałem w poprzednim numerze „Bajtki”, to na tym ten temat zakończę.

Urządzeniem sprawiającym użytkownikom najwięcej kłopotów jest drukarka. Każda chce, aby była najlepsza, najtańsza i — na dodatek — przyłączana bez interfejsu. Niestety, takich wymagań żadna drukarka nie spełnia. Zaczniemy od końca, czyli od interfejsu. Jest on zbędny przy korzystaniu z drukarek firmy Atari. W miarę nowoczesne drukarki XDM121 i XMM801 są w Polsce niedostępne. W Pewexie były kiedyś sprzedawane drukarki Atari 1029, można również napotkać gdzieś niedźwiedzie sprowadzone prywatnie 1027. Drogi użytkowniku! Kup sobie taką drukarkę, jeśli masz mało problemów, w przeciwnym razie lepiej przepisywać dane z ekranu.

Jaką drukarkę należy więc wybrać? Powinna ona spełniać dwa warunki zgodności z Atari: sprzętowy i programowy. Warunek sprzętowy polega na zastosowaniu odpowiedniego złącza, bo wiemy już, że interfejs jest niezbędny. Możliwe są dwa warianty — każda drukarka ma złącze równoległe Centronics lub szeregowe RS232. Można wykorzystać zarówno jedno, jak i drugie, ale interfejs Centronics można kupić w Polsce za około 40 dolarów, RS232 zaś kosztuje około 100 dolarów i jest w Polsce niedostępny. Warunkiem programowym jest zgodność drukarki z posiadanym oprogramowaniem, co pozwoli na pełne wykorzystanie możliwości zarówno programów, jak i sprzętu. Przeważająca większość programów dobrze współpracuje z drukarkami w standardzie Epson i taką właśnie należałoby wybrać. Może to być więc drukarka Epson, Star, Citizen lub podobna.

DUŻE PIĘKNIJSZE

Komputery serii ST reprezentują zupełnie inny, znacznie wyższy poziom techniki. Informacje na ich temat są publikowane znacznie częściej, i to nie tylko w prasie komputerowej. Mimo to znajomość tych komputerów

jest wśród naszych czytelników zwykle gorsza niż ich starszych i młodszych braci.

ST, podobnie jak „małe” Atari, różni się między sobą przede wszystkim pojemnością pamięci operacyjnej. Dodatkowo poszczególne odmiany mają nieco inną konfigurację sprzętową. Rozpoznanie tej konfiguracji jest bardzo proste, gdyż niemal wszystkie informacje są uwidocznione w nazwie komputera (choć początkowo panował w tej dziedzinie pewien bałagan). Podstawowe dane komputerów serii ST są następujące:

- procesor 68000,
- zegar taktujący 8 MHz,
- system operacyjny w pamięci ROM 192 KB,
- 3 tryby rozdzielczości obrazu,
- maksymalna rozdzielczość 640 x 400 punktów,
- wybór z palety 512 kolorów,
- 3 generatory dźwięku,
- klawiatura QWERTY — 59 klawiszy,
- 8 klawiszy do obsługi kursora,
- 18 dodatkowych klawiszy numerycznych,
- 10 klawiszy funkcyjnych,
- złącze szeregowe RS232,
- złącze równoległe Centronics,
- złącze DMA dla twardego dysku i drukarki laserowej,
- złącze zewnętrznej stacji dysków,
- złącze MIDI dla instrumentów muzycznych,
- dwa gniazda joysticków,
- gniazdo modułu ROM (cartridge'a),
- gniazdo monitorowe.

Pojemność pamięci operacyjnej ST jest określona przez nazwę (różnice pomiędzy liczbą w nazwie i liczbą kilobajtów wynikają z przeliczenia bajtów): 260 to 256 KB (262144 bajty), 520 — 512 KB (524288 B), 1040 — 1024 KB (1048576 B), a MEGA 2 i MEGA 4 to odpowiednio 2 lub 4 MB. Dodatkowe litery nazwy określają konfigurację komputera. Litera „M” oznacza wbudowany modulator telewizyjny PAL — zewnętrznie można to rozpoznać po dodatkowym gnieździe antenowym. Litera „F” symbolizuje natomiast wbudowaną dwustronną stację dysków (w starszych egzemplarzach jest to stacja jednostronna). Litery tej nie mają tylko modele MEGA, gdyż wbudowana stacja jest ich wyposażeniem standardowym. Ten przejrzysty schemat był jednak zakłócony w początkowej fazie rozwoju komputerów ST, gdyż 512 KB pamięci oznaczano niekiedy jako 260+, a 1024 KB — jako 520+.

Rozwój Atari ST odbywał się na naszych oczach, już po wprowadzeniu tych komputerów do sprzedaży. Najistotniejszym aspektem tego rozwoju są zmiany w systemie operacyjnym komputera, które mogą powodować, że niektóre programy nie będą działały. Początkowo system operacyjny (TOS) był bowiem odczytywany z dyskietki. Sposób ten był jednak bardzo kłopotliwy i szybko TOS został zapisany w pamięci ROM. W trakcie eksploatacji wykryto w TOS-ie kilka błędów oraz niewygodę pewnych rozwiązań, co spowodowało konieczność wprowadzenia zmian. Rozpoznanie wersji systemu operacyjnego jest możliwe po uruchomieniu komputera. Po wybraniu z menu wariantu DESKTOP INFO wyświetlana jest informacja o TOS-ie. Stara wersja ma datę 1985, nowa zaś 1986—1987. Jeszcze nowsza wersja, zajmująca powiększoną pamięć ROM, została uwzględniona przez dodanie do nazwy litery „E” (gdyż zmiany obejmują nie tylko system operacyjny).

Obecnie są produkowane tylko komputery ST z nową wersją systemu oraz STE, nie ma tu więc kłopotów z wyborem. Czytelnicy kupujący sprzęt używany pytają jednak, która wersja TOS-u jest lepsza. Łatwo zauważyć, że pytanie takie jest bez sensu — gdyby stary system był lepszy, to nie zostałby zmieniony.

MONITORY

Do komputerów ST przeznaczone są specjalne monitory Atari, które występują w dwóch odmianach — kolorowej (SC) i monochromatycznej (SM). Monitory SC pozwalają na pracę tylko w niskiej i średniej rozdzielczości, a SM wyłącznie w wysokiej. Dużą część programów działa niezależnie od rodzaju zastosowanego monitora, ale wiele jest przeznaczonych dla konkretnego typu — zwykle gry wymagają koloru, a programy użytkowe nie. Najlepszym — lecz droгим rozwiązaniem — jest posiadanie obu monitorów i wykorzystywanie ich odpowiednio do potrzeb. Gdy niedostatek pieniędzy nie pozwala na zakup dwóch monitorów, należy dokonać wyboru w zależności od przewidywanego zakresu zastosowania komputera. Niezdecydowanym polecam nabyć monitora monochromatycznego i komputera z modulatorem do telewizora.

Pozostaje jeszcze problem przyłączenia innego monitora (nie Atari). Jest to możliwe, lecz wymaga własnoręcznego wykonania odpowiedniej wtyczki, gdyż zastosowane w komputerze gniazdo jest nietypowe.

STACJE DYSKÓW

Jedynymi pamięciami masowymi stosowanymi w Atari ST są stacje dysków elastycznych i twarde dyski. Uwaga ta jest konieczna, gdyż niektórzy czytelnicy pytają o możliwość współpracy ST z magnetofonem. No cóż, można zaprząć konia do ferrari, tylko po co?

Firma Atari produkuje dwa typy stacji do ST: dwustronne SF314 i jednostronne SF354. Użytkowanie stacji jednostronnych jest niewygodne i kosztowne — na każdej dyskietce mieści się dwukrotnie mniej informacji (360 KB wobec 720 na dwustronnej), bo zapisywana jest tylko jedna strona, a dyskietki takiej (3 1/2 cala) nie można odwrócić. Przy zakupie stacji należy więc uważać, aby nie nabrać się na „wyjątkową” okazję. Pozostałe dane techniczne stacji Atari są następujące:

- dyskietki o średnicy 3 1/2 cala,
- transmisja szeregowa,
- prędkość transmisji 128000 bodów.

Stacje do Atari są produkowane również przez inne firmy (m.in. Cumana i California Access). Jeśli są to stacje do ST, to nie ma żadnego problemu. Wykorzystanie stacji przeznaczonych dla komputerów innych firm może sprawić nieco kłopotu; w najlepszym razie będzie to konieczność wymiany wtyczki, a w najgorszym — kontrolera stacji (wtedy lepiej kupić inną stację).

Bardzo podobnie jest z twardymi dyskami. Najpierw Atari produkowało stacje SH317 o pojemności 10 MB oraz stacje SH204 i SH205 o pojemności 20 MB. W związku z rozwojem techniki zamiast nich obecnie są w produkcji stacje MEGAFIL 30 i MEGAFIL 60, których pojemność jest określona przez liczbę w nazwie: 30 lub 60 MB. Z produktów innych firm najbardziej znane są stacje Supra, których pojemności wynoszą 20, 30, 40 lub 60 MB. Ponadto do pracy z ST można przystosować niemal każdy napęd twardego dysku, który posiada kontroler SCSI. Konieczne może być jednak wykonanie odpowiedniego interfejsu. Oto najważniejsze dane twardych dysków Atari:

- kontroler standardu SCSI,
- transmisja równoległa,
- prędkość transmisji 1.330.000 bodów.

Ponadto możliwe jest dołączenie do ST stacji dysków dla dyskietek o średnicy 5 1/4 cala. Stacje takie umożliwiają m.in. bezpośrednie przenoszenie danych pomiędzy ST i komputerami klasy IBM. Pozwalają one również na pewne oszczędności finansowe, gdyż stosowane w nich dyskietki są tańsze. Stacje tego typu są wykonywane także w Polsce.

Wojciech Zientara

BZZZZZ

W „Bajtku” 3/87
została wydrukowana
gra „Bzzzzzz” w wersji
dla ZX Spectrum.
Przetłumaczyłem ją na
Atari i w tej postaci
chciałbym
zaprezentować
czytelnikom.
Zasady tej gry zostały
opisane w „Bajtku”
3/87. Ponadto są one
niemal identyczne, jak
zasady „Gry w linie”
opublikowanej
w „Moim Atari”
numer 1.
Program ten jest
krótki i prosty, więc
każdy może
rozbudować go według
własnego uznania.

*Marek
Frueauff*



```
VC 1 REM *** Bzzzzz ***
FB 2 REM Marek Frueauff
DH 3 REM (c) 1990, Sp. Bajtek
NJ 4 REM
PW 10 DIM A$(6),B$(6),QA$(1)
OB 20 ? CHR$(125):POKE 710,193:POKE 755,0
MU 30 POSITION 10,10: ? "DO ILU GRACIE"
OH 40 TRAP 40:INPUT PKT:TRAP 40000: ?
KQ 50 POSITION 10,10: ? "PODAJCIE IMIONA"
VQ 60 ? "GRACZ 1":INPUT A$
XA 70 ? "GRACZ 2":INPUT B$
WE 100 S=INT(RND(0)*4+1):H=INT(RND(0)*4+1)
JQ 110 GRAPHICS 7:COLOR 2:POKE 710,198:POKE 755,0
ZO 120 FOR I=5 TO 153:PL0T I,3:PL0T I,77:
YN 130 FOR I=3 TO 77:PL0T 153,I:PL0T 4,I:
XM 140 ? A$;" PUNKT=";PUNKT1;" ";B$;"
LC 150 A=STICK(0):B=STICK(1)
SK 160 IF A<7 OR A=9 OR A=10 OR A=15 THEN
NL 170 IF A=14 THEN R=R-1:S=1
MM 180 IF A=13 THEN R=R+1:S=2
VI 190 IF A=11 THEN E=E-1:S=3
SO 200 IF A=7 THEN E=E+1:S=4
OH 210 GOTO 260
UR 220 IF S=1 THEN R=R-1
TP 230 IF S=2 THEN R=R+1
DP 240 IF S=3 THEN E=E-1
CN 250 IF S=4 THEN E=E+1
WW 260 IF B<7 OR B=9 OR B=10 OR B=15 THEN
KY 270 IF B=14 THEN W=W-1:H=1
JZ 280 IF B=13 THEN W=W+1:H=2
DB 290 IF B=11 THEN Q=Q-1:H=3
ZU 300 IF B=7 THEN Q=Q+1:H=4
OS 310 GOTO 360
YG 320 IF H=1 THEN W=W-1
XE 330 IF H=2 THEN W=W+1
QW 340 IF H=3 THEN Q=Q-1
PU 350 IF H=4 THEN Q=Q+1
FN 360 LOCATE Q,W,Z:IF Z=2 THEN 500
VR 370 LOCATE E,R,X:IF X=2 THEN 600
QZ 380 PL0T Q,W:PL0T E,R:GOTO 150
XM 400 FOR I=0 TO 100:POKE 710,193
JV 410 SOUND 3,1,4,15:SETCOLOR 1,2,1
FY 420 NEXT I
DY 430 SOUND 3,0,0,0:SETCOLOR 1,2,10
ZI 440 RETURN
PA 500 PUNKT1=PUNKT1+1
PH 510 ? "
SB 520 GOSUB 400
VY 530 IF PKT<>PUNKT1 THEN 100
TM 540 GOSUB 700
VV 550 ? #6;" ";A$
OO 560 GOTO 800
QC 600 PUNKT2=PUNKT2+1
XL 610 ? "-----"
SC 620 GOSUB 400
WR 630 IF PKT<>PUNKT2 THEN 100
TN 640 GOSUB 700
WM 650 ? #6;" ";B$
OP 660 GOTO 800
KD 700 GRAPHICS 2:POKE 710,193:POKE 755,0
SO 710 ? #6;" BRAWO BRAWO BRAWO"
PX 720 FOR I=1 TO 6:READ QA$
WQ 730 FOR SPOW=0 TO 60:NEXT SPOW
ZG 740 ? #6: ? #6;" ";QA$:
MX 750 NEXT I:RETURN
TS 800 FOR I=0 TO 400:SETCOLOR 0,2,1:NEXT I
JE 810 GRAPHICS 0:POKE 710,193:POKE 755,0
LY 820 POSITION 10,10: ? "NACISNIJ RETURN"
FZ 830 SETCOLOR 1,2,10:INPUT QA$:RUN
SR 1000 DATA W,Y,G,R,A,L
```


PERMUTACJE

Wykorzystuję komputer — między innymi — do nauki, chciałbym przedstawić Czytelnikom jeden z programów, który w tym celu napisałem. Dotyczy on działu matematyki zwanego kombinatoryką, a dokładniej — zagadnienia znanego pod nazwą permutacji.

Według bardzo mądrych definicji permutacją bez powtórzeń zbioru y złożonego z n różnych elementów nazywamy każdą funkcję różnowartościową określoną na zbiorze $1, 2, \dots, n$ i przyjmującą wartości ze zbioru y . W praktyce można powiedzieć, że zbiorem wszystkich permutacji zbioru y (ciągu y) jest zbiór wszystkich możliwych ustawień elementów zbioru y . Na przykład, gdy ciąg y zawiera elementy A, B i C, to zbiór wszystkich permutacji zbioru y zawiera elementy: ABC, ACB, BAC, BCA, CAB i CBA. W definicji jest podane, że elementy zbioru y nie mogą się powtarzać. Mój program nie stawia takiego warunku, gdyż traktuje każdy element zbioru y jako inny, np. permutacjami ciągu AA są ciągi AA i AA.

Program powstał pierwotnie w języku Turbo Basic XL (listing 1). W szkole mam dostęp do komputera MSX, więc napisałem też wersję tego programu w języku MSX Basic. Działa on jednak bardzo wolno na obu komputerach. Powstała więc wersja w języku maszynowym, która działa o wiele szybciej. Program maszynowy został umieszczony w instrukcjach DATA uruchamiającego go programu w Basicu (listing 2). Dodatkowo posiada on licznik permutacji.

Poza czysto teoretycznym zastosowaniem do nauki, program ten może przydać się w praktyce szaradziom, maniakom krzyżówek itp. Podaje on bowiem wszystkie anagramy danego słowa. Jeżeli w wierszu 10 drugiego programu w instrukcji OPEN zamiast „E:” poda się „P:”, to wyniki działania programu zostaną utrwalone na drukarce. Wypisywanie permutacji można zatrzymać kombinacją klawiszy <CONTROL><1>.

Marcin Brzeziński

P.S. Listing 3 jest wersją źródłową procedury maszynowej zawartej w listingu 2. Jest on przeznaczony dla osób interesujących się programowaniem w języku maszynowym. Nie trzeba go wpisywać dla uzyskania poprawnego działania pozostałych programów.

Osoby zainteresowane efektywnym i eleganckim algorytmem generującym permutacje zbioru n -elementowego odsyłamy do artykułu Algebrak-Metody czołgowe („BAJTEK” 3—4/90).

(red.)

LISTING 1

```
LW 1 REM PERMUTACJE
KN 2 REM Marcin Brzezinski
DH 3 REM (c) 1990, Sp. Bajtek
NJ 4 REM
LD 10 DIM C$(100),ROB$(200),PER$(100),POM
$(99),STIN(100):PP=%1:X=%1:FOR I=%1 TO
100:STIN(I)=%1:NEXT I
BD 20 # A:?:? "Wydruk na drukarce ? (T/N
)":GET KEY:ON KEY<>78 AND KEY<>84 GO#
A:DK=(KEY=84):?:?
GD 30 ? "Podaj ciąg znakow do permutacji
(max. 100)":INPUT C$:ROB$=C$
JA 40 # B:IF X>LEN(ROB$) AND PP=%1 THEN R
UN
UI 50 IF X>LEN(ROB$) THEN EXEC USTUP:IF X
<=LEN(ROB$):EXEC USTRIGHT:GO# B:ELSE :
GO# B:ENDIF
JY 60 # C:PER$(PP,PP)=ROB$(X,X):IF PP=LEN
(C$) THEN EXEC DRUK:X=X+%1:GO# B
JP 70 EXEC USTDOWN:GO# C
OO 100 -----
US 110 PROC USTDOWN
FD 120 IF X=LEN(ROB$) THEN ROB$=ROB$(%1,X
-%1):GO# D
RI 130 ROB$(X)=ROB$(X+%1)
IU 140 # D:PP=PP+%1:X=STIN(PP)
VT 150 ENDPROC
OP 200 -----
JP 210 PROC USTUP
AC 220 PP=PP-%1:X=STIN(PP):IF X>LEN(ROB$)
THEN ROB$(X,X)=PER$(PP,PP):GO# E
CQ 230 POM$=ROB$(X):ROB$(X+%1)=POM$:ROB$(
X,X)=PER$(PP,PP)
WS 240 # E:ENDPROC
OQ 300 -----
OO 310 PROC USTRIGHT
LB 320 X=X+%1:STIN(PP)=X:FOR I=LEN(C$) TO
PP+%1 STEP -%1:STIN(I)=%1:NEXT I
VR 330 ENDPROC
OR 400 -----
HC 410 PROC DRUK
IU 420 ? PER$:IF DK THEN LPRINT PER$
VS 430 ENDPROC
```

LISTING 2

```
LW 1 REM PERMUTACJE
KN 2 REM Marcin Brzezinski
DH 3 REM (c) 1990, Sp. Bajtek
NJ 4 REM
ZH 10 OPEN #1,8,0,"E:"
PS 12 ? "Proszę chwile poczekać...":CHR$(
160):CHR$(30):
DT 15 DIM C$(100):GOSUB 400
MJ 20 ? CHR$(125):"*****
***** PERMUTACJE v2.1
LUBLIN **"
FE 25 ? "Marcin Brzezinski 1.02.89
*****"
MW 30 ? "Podaj ciąg znakow do permutacji"
:INPUT C$
CF 40 A=USR(30313,ADR(C$),LEN(C$))
RD 50 ? :? "Wciśnij dowolny klawisz":CHR$(
160):CHR$(30):
FN 60 OPEN #5,4,0,"K:":GET #5,A:CLOSE #5:
GOTO 20
SM 400 RESTORE 500:AD=30313:S=0:TRAP 420
VQ 410 READ D:POKE AD,D:S=S+D:AD=AD+1:GOT
O 410
TA 420 IF D>S THEN ? :? "Błąd w wierszac
h DATA":END
ZG 430 RETURN
LP 500 DATA 216,104,104,133,205,104,133,2
04,104,104,208,1,96,141,3,6,160,12,169
,32,153,247,117,136,208
TM 510 DATA 250,169,0,141,0,6,141,1,6,169
,1,141,4,6,141,5,6,172,3,6,153,147,117
,136,208
CP 520 DATA 250,177,204,153,48,117,200,20
4,3,6,208,245,140,0,6,169,11,141,82,3,
169,248,141,84,3
WQ 530 DATA 169,117,141,85,3,152,24,105,1
3,141,88,3,170,169,155,157,247,117,169
,0,141,89,3,173,4
XR 540 DATA 6,205,0,6,144,32,240,30,173,5
,6,201,1,208,1,96,32,76,119,173,4,6,20
5,0,6
EH 550 DATA 240,5,144,3,76,203,118,32,143
,119,76,203,118,172,5,6,174,4,6,189,47
,117,153,3,118
CX 560 DATA 204,1,6,144,5,240,3,140,1,6,2
04,3,6,208,9,32,167,119,238,4,6,76,203
,118,32
```

```
VU 570 DATA 29,119,76,243,118,138,205,0,6
,240,25,238,4,6,172,4,6,206,4,6,185,47
,117,157,47
TF 580 DATA 117,204,0,6,240,5,200,232,76,
44,119,206,0,6,238,5,6,172,5,6,185,147
,117,141,4
VB 590 DATA 6,96,206,5,6,172,5,6,185,147,
117,141,4,6,205,0,6,144,15,240,13,170,
185,3,118
HM 600 DATA 157,47,117,238,0,6,76,142,119
,174,0,6,238,0,6,172,0,6,189,47,117,15
3,47,117,236
SD 610 DATA 4,6,240,5,202,136,76,117,119,
172,5,6,185,3,118,157,47,117,96,238,4,
6,172,5,6
GO 620 DATA 173,4,6,153,147,117,169,1,200
,153,147,117,204,3,6,208,247,96,160,10
,185,247,117,24,105
DH 630 DATA 1,201,33,208,5,169,49,76,196,
119,201,58,208,8,169,48,153,247,117,13
6,208,229,153,247,117
WA 640 DATA 162,16,32,86,228,96,36207
```

LISTING 3

```
O100 ;PERMUTACJE
O110 ;Marcin Brzezinski
O120 ;(c) 1990, Sp. Bajtek
O130 ;
O140 C = 204
O150 IOCB = 832
O160 CIO = 58454
O170 ROB = 29999
O180 STIN = 30099
O190 LICZNIK = 30200
O200 PER = 30211
O210 DLROB = 1536
O220 DLPER = 1537
O230 DLC = 1539
O240 X = 1540
O250 PP = 1541
O260 ;
O270 *= 30313
O280 ;
O290 CLD
O300 PLA ;pobierz ze
O310 PLA ;stosu adres
O320 STA C+1 ;i dlugosc C$
O330 PLA
O340 STA C
O350 PLA
O360 PLA
O370 BNE ZERO
O380 RTS
O390 ZERO STA DLC
O400 LDY #12 ;inicjacja
O410 LDA #32
O420 SKO STA LICZNIK-1,Y
O430 DEY
O440 BNE SKO
O450 LDA #0
O460 STA DLROB
O470 STA DLPER
O480 LDA #1
O490 STA X
O500 STA PP
O510 LDY DLC
O520 SK1 STA STIN,Y
O530 DEY
O540 BNE SK1
O550 SK2 LDA (C),Y ;przepisz C$
O560 STA ROB+1,Y ;do ROB
O570 INY
O580 CPY DLC
O590 BNE SK2
O600 STY DLROB
O610 LDA #11 ;wpisz
O620 STA IOCB+16+2 ;odpowiednie
O630 LDA # <LICZNIK ;wartosci
O640 STA IOCB+16+4 ;do IOCB
O650 LDA # >LICZNIK
O660 STA IOCB+16+4
O670 TYA
O680 CLC
O690 ADC #13
O700 STA IOCB+16+8
O710 TAX
O720 LDA #155
O730 STA LICZNIK-1,X
O740 LDA #0
O750 STA IOCB+16+9
O760 ;
O770 PTG LDA X ;petla glowna
O780 CMP DLROB
O790 BCC SK5
O800 BEQ SK5
O810 LDA PP ;jesli X>DLROB
O820 CMP #1 ;oraz PP=1, to
O830 BNE SK3 ;koniec
O840 RTS ;programu
O850 SK3 JSR USTUP
O860 LDA X
```



```

0870      CMP DLROB
0880      BEQ SK4
0890      BCC SK4
0900      JMP PTG
0910 SK4   JSR USTRIGHT
0920      JMP PTG
0930 SK5   LDY PP
0940      LDX X
0950      LDA ROB,X
0960      STA PER,Y
0970      CPY DLPER
0980      BCC SK6
0990      BEQ SK6
1000      STY DLPER
1010 SK6   CPY DLC
1020      BNE SK7
1030      JSR DRUK
1040      INC X
1050      JMP PTG
1060 SK7   JSR USTDOWN
1070      JMP SK5
1080      ;
1090 USTDOWN TXA      ;podprogram
1100      CMP DLROB   ;likwiduje
1110      BEQ SP2      ;pozycje X w ROB
1120      INC X        ;(przesuwa ROB
1130      LDY X        ;od pozycji X+1
1140      DEC X        ;do DLROB
1150 SP1   LDA ROB,Y  ;o jedna pozycje
1160      STA ROB,X    ;w prawo
1170      CPY DLROB
1180      BEQ SP2
1190      INY
1200      INX
1210      JMP SP1
1220 SP2   DEC DLROB
1230      INC PP
1240      LDY PP
1250      LDA STIN,Y
1260      STA X
1270      RTS
1280      ;
1290 USTUP DEC PP      ;podprogram
1300      LDY PP        ;rozsuwa ROB
1310      LDA STIN,Y    ;w prawo od
1320      STA X        ;pozycji X
1330      CMP DLROB    ;o jedno miejsce
1340      BCC SP3      ;i wstawia w ROB
1350      BEQ SP3      ;na pozycji X
1360      TAX          ;PER z pozycji
1370      LDA PER,Y    ;PP
1380      STA ROB,X
1390      INC DLROB
1400      JMP SP6
1410 SP3   LDX DLROB
1420      INC DLROB
1430      LDY DLROB
1440 SP4   LDA ROB,X
1450      STA ROB,Y
1460      CPX X
1470      BEQ SP5
1480      DEX
1490      DEY
1500      JMP SP4
1510 SP5   LDY PP
1520      LDA PER,Y
1530      STA ROB,X
1540 SP6   RTS
1550      ;
1560 USTRIGHT INC X    ;podprogram
1570      LDY PP        ;dodaje 1 do X,
1580      LDA X          ;zapamietuje X
1590      STA STIN,Y    ;w STIN o
1600      LDA #1        ;pozycji PP.
1610 SP7   INY          ;a nastepnie
1620      STA STIN,Y    ;zapelnia STIN
1630      CPY DLC      ;jedynekami od
1640      BNE SP7      ;pozycji PP+1
1650      RTS          ;do DLC
1660      ;
1670 DRUK LDY #10      ;zwieksz licznik
1680 SP8   LDA LICZNIK-1,Y ;o 1
1690      CLC          ;i drukuj
1700      ADC #1        ;licznik oraz
1710      CMP #33       ;PER na ekranie
1720      BNE SP9
1730      LDA #49
1740      JMP SP10
1750 SP9   CMP #58
1760      BNE SP10
1770      LDA #48
1780      STA LICZNIK-1,Y
1790      DEY
1800      BNE SP8
1810 SP10  STA LICZNIK-1,Y
1820      LDX #16
1830      JSR CIO
1840      RTS

```

ERRATA

W „Bajtku” III (9—10/90) grasował ogromnie żarłoczny chochlik. Jego łupem padły aż trzy z zamieszczonych w klanie Atari programów. Przepraszamy Czytelników za te niespodzianki i publikujemy ponownie kompletny wydruk programu „Toto-Lotek”, procedury umożliwiające uzyskanie polskich liter w Action! oraz poprawną postać wiersza wyświetlającego dane z „Mini-edytora duszków”.

(red.)

TOTO-LOTEK

```

NV      1 REM TOTO-LOTEK
CV      2 REM LESZEK STROZOWSKI
XO      3 REM COPYRIGHT (C) BAJTEK
NJ      4 REM
CU      10 GRAPHICS 0:POKE 566,158:POKE 731,1
PD      20 POKE 752,1:EXEC POL:POKE 756,156
PQ      70 EXEC PLANSZA:DIM K(8),T(8),P*(18):P
        $=" "
YC      75 EXEC TYT
BR      80 FOR R=1 TO 7:K(R)=0:T(R)=0:NEXT R:I
        T=0
OX      90 EXEC KUPON
ZV      95 COLOR 2
EN      100 FOR P=1 TO 15
VZ      110 IF P<7 THEN TEXT 20,183,P:TEXT 28,
        183," LICZBA ?"
OY      115 IF P=7 AND 15=7 THEN TEXT 20,183,"
        DODATKOWA?"
WN      120 TEXT 120,183," " :GOSUB 240
QS      130 IF KEY=155 OR KEY=126 OR KEY=48 TH
        EN 120
VN      135 SOUND 0,50,10,10:PAUSE 3:SOUND
VD      140 K=VAL(CHR$(KEY))
AB      150 TEXT 120,183,K
TF      160 GOSUB 240
VT      165 SOUND 0,50,10,10:PAUSE 3:SOUND
ZJ      170 IF KEY=126 THEN TEXT 120,183," " :G
        OT0 120
RH      180 IF KEY=155 THEN 260
RF      190 K2=VAL(CHR$(KEY))
NY      200 TEXT 128,183,K2
GO      210 GET KEY:IF KEY<>155 AND KEY<>126 T
        HEN 210
VK      215 SOUND 0,50,10,10:PAUSE 3:SOUND
OK      220 IF KEY=126 THEN TEXT 128,183," " :G
        OT0 160
RU      230 IF KEY=155 THEN 270
BH      240 GET KEY:IF KEY<48 OR KEY>57 AND KE
        Y<>155 AND KEY<>126 THEN 240
ZI      250 RETURN
WZ      260 NR=K:GOTO 280
UA      270 NR=K*10+K2
KV      280 IF NR>ML THEN 120
FY      290 FOR R=1 TO 15
GR      295 IF NR=T(R) THEN POP :GOTO 120
JF      300 NEXT R
FO      310 T(P)=NR
FR      320 EXEC KRZYZYK
IR      330 NEXT P
GB      335 COLOR 2:TEXT 12,183," LOSOWANIE
        ! " :PAUSE 50
EZ      350 FOR P=1 TO 15
VP      360 NR=1+INT(RND*ML)
FV      370 FOR R=1 TO 15
KB      375 IF NR=K(R) THEN POP :GOTO 360
JV      380 NEXT R
CJ      390 K(P)=NR:EXEC KOLKO:PAUSE 20
IM      400 NEXT P
ES      410 FOR P=1 TO 15
FM      420 FOR R=1 TO 15
ZE      430 IF T(R)=K(P) THEN IT=IT+1
JO      440 NEXT R
JJ      445 NEXT P
PZ      450 COLOR 2:TEXT 34,183," TRAFIED: " :T
        EXT 114,183,IT:POKE 19,0
RU      452 IF IT<3 THEN EXEC S03
EJ      454 IF IT>=3 THEN EXEC S04
VY      460 DO
ET      470 IF PEEK(53279)=6 THEN 500
CO      480 IF PEEK(19)=6 THEN TEXT 8,82,P$:TE
        XT 8,90,"START - NOWY KUPON":TEXT 8,98
        ,P$

```

```

RP      490 LOOP
JW      500 POP :GOTO 75
BG      1000 PROC KRZYZYK
TF      1010 EXEC ZAL
XJ      1020 X=7+X*16:Y=-21+Y*24
RF      1025 IF P=7 THEN COLOR 1
KK      1027 EXEC S01
TM      1030 PLOT X,Y:DRAWTO X+16,Y+24
VJ      1040 PLOT X+16,Y:DRAWTO X,Y+24
YR      1050 ENDPROC
FO      1500 PROC KOLKO
TP      1510 EXEC ZAL
FV      1520 X=15+X*16:Y=-9+Y*24
KZ      1525 EXEC S02
EB      1530 COLOR 3:CIRCLE X,Y,10,16
YY      1540 ENDPROC
XD      2000 PROC ZAL
FO      2010 IF NR<=7 THEN 2100
KH      2020 Y=NR
PP      2030 REPEAT
WG      2040 Y=Y-7
JL      2050 UNTIL Y<=7
PY      2060 REPEAT
RG      2070 X=(NR-Y)/7+1
JI      2080 UNTIL X<=7
ZE      2090 ENDPROC
SI      2100 Y=NR:X=1
YI      2110 ENDPROC
LR      5000 PROC KUPON
TV      5010 GRAPHICS 31:POKE 559,0:SETCOLOR 4
        ,7,4:SETCOLOR 0,0,14:SETCOLOR 1,0,0:SE
        TCOLOR 2,3,2:COLOR 1:POKE 756,156
NR      5020 FOR R=23 TO 145-PL STEP 16
EJ      5030 PLOT R,3:DRAWTO R,170
JA      5040 NEXT R
RY      5050 FOR R=3 TO 178 STEP 24
KZ      5060 PLOT 23,R:DRAWTO 135-PL,R
JJ      5070 NEXT R
LJ      5080 FOR R=12 TO 158 STEP 24:TEXT 28,R
        ,(R-12)/24+1:NEXT R:TEXT 44,12,"8":TEX
        T 44,36,"9"
CJ      5090 FOR R=60 TO 168 STEP 24:TEXT 40,R
        ,"1":TEXT 47,R,(R-60)/24:NEXT R
CR      5100 FOR R=12 TO 120 STEP 24:TEXT 56,R
        ,"1":TEXT 63,R,(R-12)/24+5:NEXT R
HI      5110 FOR R=132 TO 160 STEP 24:TEXT 56,
        R,"2":TEXT 63,R,(R-132)/24:NEXT R
XX      5120 FOR R=12 TO 158 STEP 24:TEXT 72,R
        ,"2":TEXT 79,R,(R-12)/24+2:NEXT R:TEXT
        88,12,"2":TEXT 95,12,"9"
JJ      5130 FOR R=36 TO 168 STEP 24:TEXT 88,R
        ,"3":TEXT 95,R,(R-36)/24:NEXT R
ZX      5140 FOR R=12 TO 96 STEP 24:TEXT 104,R
        ,"3":TEXT 111,R,(R-12)/24+6:NEXT R
IS      5150 FOR R=108 TO 158 STEP 24:TEXT 104
        ,R,"4":TEXT 111,R,(R-108)/24:NEXT R
ZR      5155 IF PL THEN 5165
GJ      5160 FOR R=12 TO 168 STEP 24:TEXT 120,
        R,"4":TEXT 127,R,(R-12)/24+3:NEXT R
ZW      5165 POKE 559,34
ZD      5170 ENDPROC
QK      10000 PROC PLANSZA
EL      10010 GRAPHICS 18:SETCOLOR 0,0,0:SETCO
        LOR 3,0,0:POKE 756,156
TK      10020 POSITION 9,3: ? #6;"I\":POSITION
        9,4: ? #6;"J\":POSITION 6,6: ? #6;"SOFTW
        ARE"
TR      10030 DL=DPEEK(560):POKE DL+13,6:POSIT
        ION 5,8: ? #6;"PREZENTUJE"
SR      10040 PAUSE 50:FOR R=0 TO 15:PAUSE 4:S
        ETCOLOR 0,0,R:SOUND 0,R+5,12,R:NEXT R:
        PAUSE 8:DSOUND :PAUSE 150
ZD      10050 FOR R=15 TO 0 STEP -1:PAUSE 4:SE
        TCOLOR 0,0,R:SOUND 0,R+5,12,R:NEXT R:P
        AUSE 4:DSOUND :PAUSE 50
DP      10060 ENDPROC
SX      10490 PROC TYT
NW      10500 GRAPHICS 5:SETCOLOR 4,0,6:SETCOL
        OR 0,12,2:SETCOLOR 1,3,2:SETCOLOR 2,0,
        6:COLOR 1:POKE 756,156
BM      10505 DL=48026:POKE DL,6:POKE DL+1,6:P
        OKE DL+2,6
VH      10510 TEXT 12,0,"T O T O":TEXT 4,8,"L
        O T E K": ? : ? "SELECT - 7 Z 49": ? "OPT
        ION - 5 Z 42":COLOR 2
AQ      10520 DO
EA      10530 XL=INT(RND*64):YL=16+INT(RND*16)
        :L=1+INT(RND*49)
KA      10540 TEXT XL,YL,L:SOUND 0,L+20,10,10:
        PAUSE 5:SOUND :TEXT XL,YL," " :PAUSE 3
HW      10545 IF PEEK(53279)=5 THEN GOSUB 1100
        0:GOTO 10570
MM      10550 IF PEEK(53279)=3 THEN GOSUB 1150
        0:GOTO 10570
JN      10560 LOOP
HH      10570 POP :ENDPROC
HL      11000 IS=7:ML=49:PL=0:RETURN
SD      11500 IS=5:ML=42:PL=16:RETURN
GF      14000 PROC POL
OM      14010 MOVE 57344,39936,1024:RESTORE 15
        000:FOR R=0 TO 31:READ Q:POKE 40408+R,
        Q:NEXT R
VO      14020 RESTORE 15100:FOR R=0 TO 31:READ
        Q:POKE 40712+R,Q:NEXT R
DL      14030 ENDPROC
IN      15000 DATA 227,227,227,227,227,227,227
        ,227
LO      15010 DATA 255,255,128,128,128,128,128
        ,255
WR      15020 DATA 227,224,224,224,224,224,255
        ,255
RY      15030 DATA 255,7,7,7,7,7,255,255
        ,Y
YJ      15100 DATA 0,24,60,102,102,126,102,3
QS      15110 DATA 24,60,102,102,102,102,60,0
QX      15120 DATA 24,126,12,24,48,96,126,0
NY      15130 DATA 24,102,118,126,126,110,102,
        0
SU      20000 PROC S01
OL      20010 FOR R=1 TO 5
PE      20020 SOUND 0,50,10,12:PAUSE 1:SOUND :
        PAUSE 1
KA      20030 NEXT R

```



```
DI 20040 ENDPROC
TO 20200 PROC S02
OR 20210 FOR R=1 TO 5
PD 20220 SOUND 0,20,12,12:PAUSE 1:SOUND :
    PAUSE 1
KG 20230 NEXT R
DO 20240 ENDPROC
UI 20400 PROC S03
HO 20410 FOR R=15 TO 1 STEP -1
WW 20420 FOR H=1 TO 45
TE 20430 SOUND 0,50-H,10,R
XB 20440 NEXT H:NEXT R:SOUND
DY 20450 ENDPROC
VC 20600 PROC S04
HK 20610 RESTORE 20620:FOR R=1 TO 8:READ
    H:DSOUND 0,H,10,10:PAUSE 7:NEXT R
LM 20620 DATA 8128,6088,6834,8128,10852,9
    125,8612,8128
DW 20630 ENDPROC
```

Polskie litery w Action!

```
; Polskie litery w Action!
; Grzegorz Sarnecki
; Copyright (c) Bajtek

; tylko male litery

PROC PolLit1()
CARD n=[57344],m=[32768],a,b

    MoveBlock(m,n,1024)
    a=m+776 b=m+520
    MoveBlock(b,a,208)
    a=m+976 b=m+512
    MoveBlock(b,a,8)
    Poke(m+513,12) Poke(m+514,24)
    Poke(m+515,124) Poke(m+527,3)
    Poke(m+537,12) Poke(m+538,24)
    Poke(m+539,60) Poke(m+559,6)
    Poke(m+611,28) Poke(m+612,56)
    Poke(m+625,12) Poke(m+626,24)
    Poke(m+627,124) Poke(m+633,12)
    Poke(m+634,24) Poke(m+635,60)
    Poke(m+664,6) Poke(m+665,12)
    Poke(m+721,24) Poke(m+772,0)
    Poke(m+723,124)
; powtorzyc po kazdej Graphics()
    Poke(756,128)
RETURN
; pelny zestaw liter

MODULE
BYTE litery=756,ch

PROC PolLit()
BYTE f,g,j
CARD chb,i
BYTE ARRAY a=[
    'Q 0 0 0 255 255 24 24 24
    'A 0 24 60 102 102 126 102 12
    'B 24 24 24 31 31 24 24 24
    'C 12 60 102 96 96 102 60 0
    'D 0 0 60 102 126 96 60 6
    'E 0 126 96 124 96 96 126 12
    'F 12 24 126 12 24 48 126 0
    'G 12 24 0 126 12 48 126 0
    'H 0 0 0 31 31 24 24 24
    'I 0 0 0 248 248 24 24 24
    'J 24 24 24 255 255 0 0 0
    'K 24 24 24 248 248 24 24 24
    'L 0 96 120 112 224 96 126 0
    'M 12 24 0 124 102 102 102 0
    'N 24 102 118 126 126 110 102 0
    'O 12 60 102 102 102 102 60 0
    'P 12 24 0 60 102 102 60 0
    'Q 0 0 60 6 62 102 62 12
    'S 12 60 96 60 6 6 60 0
    'T 24 24 24 248 248 0 0 0
    'U 24 24 24 31 31 0 0 0
    'V 12 24 0 60 96 96 60 0
    'W 12 24 62 96 60 6 124 0
    'X 0 24 0 126 12 48 126 0
    'Z 24 0 126 12 24 48 126 0
    $7B 0 56 24 28 56 24 60 0]

ch=Peek(106)-8
chb=ch*256
MoveBlock(chb,57344,1024)
FOR i=0 TO 233 STEP 9 DO
    f=a(i)
    FOR j=0 TO 7 DO
        g=a(i+j+1)
        Poke(chb+f*8+j,g)
    OD
OD
; powtorzyc po kazdej Graphics()
    litery=ch
RETURN
```

Mini-edytor duszków

```
F. I=1TOLEN(A$):?ASC(
A$(X,X));",",":N. I:??
```

DWUMIAN NEWTONA

Czasem, aż dziw bierze, że właśnie — nieskomplikowane przecież — wyrażenie matematyczne przyczyniło się w tak dużym stopniu do rozwoju niemal każdej dziedziny nauk przyrodniczych, a jednocześnie tak zatrulo życie jednemu uczniowi.

Podstawową przyczyną tych kłopotów jest nie tyle sam dwumian, co jego pracochłonne rozwijanie według wzoru Newtona. Do dziś jednak jest to sposób najbardziej efektywny. Powszechnie wiadomo, że komputery można wykorzystać do obliczeń według podanych wzorów, natomiast mniej znany jest fakt, że w niektórych przypadkach można je zastosować do wyprowadzania wzorów. Pokazany obok program, napisany w Atari Basic, podaje rozwinięcie dwumianu Newtona postaci:

$(1+x)^n$ lub $(1-x)^n$ na wielomiany, ewentualnie szeregi dwumianowe dla dowolnego n. Skończone rozwinięcia uzyskuje się tylko wówczas, gdy wykładnik n jest liczbą naturalną lub zerem, w przeciwnym razie otrzymuje się nieskończone szeregi potęgowe. Dlatego właśnie, dla n ujemnych lub ułamkowych przedstawiony program może wygenerować dowolną, aczkolwiek ograniczoną liczbę początkowych wyrazów rozwinięcia według wzorów:

$(1+x)^n = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + \binom{n}{3}x^3 + \dots$

$(1-x)^n = 1 - \binom{n}{1}x + \binom{n}{2}x^2 - \binom{n}{3}x^3 + \dots$

gdzie $\binom{n}{k}$ jest symbolem Newtona

i wyraża się wzorem:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Na zakończenie chciałbym zwrócić uwagę, że posługiwanie się programem jest proste. Wszystko, co niezbędne, znajduje się w menu. Ponadto nie sędzę, aby zamieszczony program umożliwiał rozwiązanie wszelkich zagadnień związanych z dwumianami, ale w pewnych okolicznościach może on posłużyć jako pożyteczna ściągawka.

Wojciech Przybył

```
FR 0 REM Rozwijanie dwumianu Newtona
TD 1 REM Wojciech Przybył
HZ 2 REM (c) 1990, Bajtek
NI 3 REM
LW 10 DIM A$(38),B$(38):A$="(1+X)^N":B$=
    "(1-X)^N":GRAPHICS 0:POSITION 5,5:?"
    Wybierz postac dwumianu (□/□):?
    20 ? " + --> " ;A$:?" :? " - --> " ;B$
    30 ON PEEK(764)=255 GOTO 30:P=1:IF PEE
    K(764)=14 THEN P=-1:A$=B$
    PE 40 POKE 764,255:POSITION 3,7:?"
    " ;A$:?" :? " Podaj N ";:INPUT N:
    M=N:ON N=ABS(INT(N)) GOTO 70:M=10
    QS 50 ? :? " Czy 10 wyrazow rozwinięcia
    wystarcza ? (Y/N)"
    HW 60 ON PEEK(764)=255 GOTO 60:IF PEEK(76
    4)=35 THEN POKE 764,255:?" :? " Podaj l
    iczbe wyrazow ";:INPUT M:M=M+(M=N)
    DE 70 POKE 764,255:?" :? :S=P*N:K=N+1:A$(8
    )=STR$(N):J=LEN(A$)+6:A$(J-5)=" " = 1 ":
    IF N=0 THEN ? A$:END
    XY 80 A$(J)="+":A$(J+(S>0))=STR$(S):L=LEN
    (A$)+1:A$(L)="+X ":IF ABS(S)=1 THEN A$
    (J+1)="X ":L=J:IF N=1 THEN ? A$:END
    TQ 90 L=L+1:FOR I=2 TO M:S=P*S*(K-I)/I:B$
    =" "+B$(1+(S>0))=STR$(S):B$(LEN(B$)+1)
    ="*X^":IF ABS(S)=1 THEN B$(2)="X^"
    XL 100 B$(LEN(B$)+1)=STR$(I):J=LEN(B$)+1:
    B$(J)="- ":IF L+J>38 THEN ? A$:A$=" "
    :L=4:?"
    XU 110 A$(L)=B$:L=L+J:NEXT I:B$="":IF M<>
    N THEN B$=" " + "...":IF L<34 THEN A$(L)=B
    $:B$=" "
    SA 120 ? A$:?" :? " " ;B$
```

DRUKOWANIE PRZY UŻYCIU WŁASNEGO GENERATORA ZNAKÓW

Kłopoty przy drukowaniu polskich liter były przyczyną rozpowszechnienia się w naszym kraju programu **ChiWriter**, wykorzystującego tryb graficzny zarówno ekranu, jak i drukarki. Prezentowany poniżej program jest prostym zastosowaniem tego pomysłu do drukowania dowolnych plików tekstowych przy użyciu własnego, a więc modyfikowalnego, generatora znaków.

Program został napisany w języku **Turbo Pascal** (wersja 3.0) i może być wykorzystany na dowolnym komputerze, wyposażonym w kompilator tego języka (IBM, Amstrad, Spectravideo, ZX Spectrum + FDD 3000, Junior). Skompilowany program wywoływany jest z dwoma parametrami. Pierwszym jest nazwa drukowanego pliku tekstowego, drugim — nazwa pliku binarnego zawierającego generator znaków.

Sam generator ma rozmiar 2KB i zawiera 8-bajtowe definicje 256 znaków. Plik taki należy utworzyć posługując się programem do tworzenia własnych znaków o matrycy 8*8. W przypadku ZX Spectrum nadaje się do tego celu **ART STUDIO**. W przypadku innych komputerów konieczne jest skopiowanie jakiegoś generatora ekranowego o podanych parametrach. W najgorszej sytuacji należy samemu na papierze zdefiniować pożądany krój czcionki i utworzyć przy pomocy debuggera dwukilobajtowy plik binarny.

Jonasz Mayer


```

program print_file_with_user_pcg;
(*****)
(*
(*          Plik PRNFILE.PAS          (C) JM 1988
(*
(*
(* Program umożliwia wydrukowanie w trybie graficznym tekstu na
(* drukarce. Wykorzystywany jest ekranowy generator znaków zawier-
(*ający 256 wzorców. W trybie tekstowym drukarka pracująca
(* w standardzie EPSON FX80 pozwala wykorzystać tylko kody od 32
(* do 127. Wyższym kodom odpowiadają te same znaki w piśmie
(* pochyłym.
(*
(*
(*****)
const
    linelength = 120; (* liczba kolumn w wierszu na drukarce *)
    rows       = 7;   (* rows+1 = liczba wierszy macierzy znaku *)
    columns    = 7;   (* columns+1 = liczba kolumn macierzy znaku *)
    gensize    = 255; (* gensize+1 = liczba znaków w generatorze *)
    no_of_gen  = 1;   (* liczba generatorów znaków *)
    recs       = 16;  (* liczba rekordów (a 128 b) na generator *)

type
    pchar = array (0..columns) of byte; (* znak na drukarce *)
    schar = array (0..rows) of byte;   (* znak na ekran *)
    pgen = array (0..gensize) of pchar; (* generator drukarkowy *)
    sgen = array (0..gensize) of schar; (* generator ekranowy *)
    matrix = array (0..rows, 0..columns) of byte; (* macierz znaku *)
    string20 = string(20);

var
    error : boolean;
    p : pgen;
    gfile : file; (* plik generatora znaków *)
    tfile : text; (* plik tekstowy do przetworzenia *)

procedure load_pgen (filename:string20; var p : pgen;
    var error : boolean);
(*****)
(* procedura ładuje z dysku ekranowy generator znaków *)
(* (bajty macierzy znaku odpowiadają wierszom) i zamienia *)
(* go w generator drukarkowy (bajty odpowiadają kolumnom) *)
(*****)
var buffer : sgen;

procedure sgen_to_pgen (var p : pgen; var s : sgen);
(*****)
(* zamiana generatora ekranowego (sgen) na generator *)
(* drukarkowy (pgen) *)
(*****)
var M : matrix;
    i : integer;

procedure S_to_M (var M : matrix; var s : schar);
(*****)
(* zamiana 8 bajtów znaku generatora ekranowego *)
(* na macierz 8*8 *)
(*****)
var i,j,x,mask : byte;
begin
    for i := 0 to rows
    do begin
        x := s(i); mask := $80;
        for j := 0 to columns
        do begin
            if (x and mask) <> 0
            then M(i,j) := 1
            else M(i,j) := 0;
            mask := mask shr 1; (* div 2 *)
        end;
    end;
end; (* of S to M *)

procedure M_to_P (var p : pchar; var M : matrix);
(*****)
(* zamiana macierzy 8*8 na 8 bajtów znaku *)
(* drukarkowego generatora *)
(*****)
var i,j,sum,mlt : byte;
begin
    for i := 0 to columns
    do begin
        sum := 0; mlt := $80;
        for j := 0 to rows
        do begin
            if M(j,i)=1
            then sum := sum + mlt;
            mlt := mlt shr 1; (* mult 2 *)
        end;
        p(i) := sum;
    end;
end; (* of M to p *)

begin (* sgen to pgen *)
    for i := 0 to gensize
    do begin
        S_to_M (M,s(i));

```

```

        M_to_P (p(i),M);
    end;
end; (* sgen to pgen *)

begin (* load pgen *)
    assign(gfile,filename); (*I-*)
    reset (gfile); (*I+*)
    if IOresult<>0
    then error := true (* brak pliku generatora ! *)
    else begin
        error := false;
        BlockRead(gfile,buffer,recs);
        Sgen_to_pgen ( p,buffer );
    end;
end; (* load pgen *)

procedure processfile (filename : string20; var p : pgen;
    var error : boolean);
(*****)
(* przetwarzanie pliku tekstowego *)
(*****)
var ch : char;
    i : integer;
    line : array (1..linelength) of char;

procedure processline (q : integer);
(*****)
(* przetwarzanie linii tekstu *)
(*****)
var i : integer;

procedure processchar (c : char);
(*****)
(* przetwarzanie pojedynczego znaku *)
(*****)
var i : byte;
begin
    for i := 0 to columns
    do write (1st, chr ( p(ord(c),i) ));
end; (* of process character *)

begin (* process line *)
    if q>0
    then begin
        write(1st,chr(27),'L',chr(lo(8*q)),chr(hi(8*q)));
        for i := 1 to q
        do processchar(line(i));
        end;
        writeln(1st);
    end; (* process line *)

begin (* process file *)
    assign(tfile,filename); (*I-*)
    reset (tfile); (*I+*)
    if IOresult<>0
    then error := true (* brak pliku do przetwarzania *)
    else begin
        error := false;
        while not eof(tfile)
        do begin
            i := 1;
            while not eoln(tfile)
            do begin
                read(tfile,ch);
                line(i) := ch;
                i := i + 1;
            end;
            readln(tfile);
            processline(i-1);
        end;
    end;
    close(tfile);
end; (* of processfile *)

begin (* main *)
    if paramcount<2
    then writeln (' Błędne wywołanie, ',
        '(PRNFILE nazwa_pliku nazwa_generatora)')
    else begin
        load_pgen (paramstr(2),p,error);
        if error
        then writeln(' Brak pliku generatora, ')
        else begin
            processfile(paramstr(1),p,error);
            if error
            then writeln(' Brak pliku do drukowania,');
        end;
    end;
end;

(*****)

Listing VI.4. Plik PRNFILE.PAS

```


SZPERACZ DYSKOWY

Opracowanie niniejsze jest uzupełnieniem opublikowanego na łamach „KOMPUTERA” artykułu pt. „Nieśmiertelność na dyskietce”.

Jak wiadomo, stacje dysków Commodore nie należą w żadnym wypadku do stacji, które można by nazwać szybkimi. Napisanie programu w BASIC-u zajmującego się wyszukiwaniem określonej sekwencji kodów na dyskietce jest oczywiście możliwe; problem polega na tym, że trwałoby to godzinami, ze względu na bardzo wolną (bo szeregową) komunikację komputera ze stacją dysków. Przeszukiwanie całej dyskietki trwa ok. 2 godzin; kompilacja programu skraca ten czas do 40—50 minut. Mimo wszystko i tak jest to o wiele za długo, aby myśleć o normalnej pracy. A trzeba sobie zdawać sprawę z faktu, że poszukiwania „nieśmiertelności” w grach odbywają się trochę „po omacku” i szybkie sprawdzanie skutków wprowadzanych zmian staje się celem pierwszorzędym.

Jeżeli nawet użytkownik chciałby taki program napisać w języku maszynowym, to uzyskałby efekty lepsze o ok. 50% (czyli 25 minut na wyszukanie sekwencji na całej dyskietce). Problem polega tu przede wszystkim na wyeliminowaniu ciągłej komunikacji z komputerem — w myśl naszych potrzeb komunikacja taka jest pożądana wyłącznie po odnalezieniu zadanej sekwencji. Cała reszta procesu powinna się odbywać np. w pamięci RAM stacji dysków, co jest jak najbardziej możliwe.

Przedstawiony tu program działa właśnie według tego schematu. Najpierw instaluje on w pamięci RAM stacji procedurę sprawdzającą w kolejnym sektorze obecność szukanej sekwencji. W ten sposób zostaje wyeliminowana komunikacja pomiędzy komputerem i stacją; przesłanie danych do komputera następuje bowiem wyłącznie wtedy, gdy została odnaleziona poszukiwana sekwencja.

Rezultaty przeszły nasze najśmielsze oczekiwania. Przeszukiwanie całej dyskietki trwa ok. 2.5 minuty. Aby umożliwić korzystanie z tego programu wszystkim użytkownikom, podajemy jego wersję gotową do wprowadzenia w formie programu napisanego w BASIC-u. Z uwagi na dużą ilość linii zawierających DATA dołączyliśmy też odpowiednią procedurę kontrolną sprawdzającą poprawność wpisanych danych. Listing został sporządzony ze sprawdzonego i działającego programu.



Bardziej zaawansowani użytkownicy mogą oczywiście uruchomić program w BASIC-u i przenieść go w postaci maszynowej na dysk za pomocą dowolnego monitora.

W tym celu wykonaj:
S”DISK SEARCHER”,08,C000,CFEF

Na dyskietce powinno być wolnych minimum 17 bloków, gdyż tyle właśnie zajmuje nasz program w wersji maszynowej. Nic również nie stoi na przeszkodzie, aby użytkownicy nie dysponujący jeszcze własną stacją dysków zapisali ten program na taśmie — w tym celu wystarczy jedynie zmienić parametr 08 na 01.

Obsługa programu nie jest trudna, ważne jest jednak przestrzeganie podanej syntaktyki poleceń. Parametry numeryczne można wprowadzać dziesiętnie lub szesnastkowo (dzięki karcie FINAL II).

Program ten wyszukuje na dyskietce:

- tekst zapisany w postaci kodów ASCII,
- tekst zapisany za pomocą znaków (kodów) ekranowych,
- podaną sekwencję bajtów.

Szperacz może pracować w jednym z dwóch trybów: przeszukiwania całej dyskietki lub wyszukiwania dowolnej sekwencji w pojedynczym programie. A oto dostępne polecenia:

A. Tryb wyszukiwania sekwencji w programie:

- PAS ”tekst”** — wyszukaj sekwencję w kodzie ASCII,
- PSC ”tekst”** — wyszukaj sekwencję w kodzie ekranowym,
- PBY B1,B2,...B30** — wyszukaj sekwencję bajtów B1,B2,...B30
- PHC tr,se** — rozpocznij wyszukiwanie ciągłe,
- PHS tr,se** — rozpocznij wyszukiwanie i zakończ je po pierwszym odnalezieniu zadanej sekwencji,
- PCC** — kontynuuj przerwane wyszukiwanie,
- PIN** — inicjalizacja początkowa w/w/wył.,
- PDS** — wyświetl następną ścieżkę i sektor.

B. Tryb wyszukiwania na całej dyskietce:

- DAS ”tekst”** — wyszukaj sekwencję w kodzie ASCII,
- DSC ”tekst”** — wyszukaj sekwencję w kodzie ekranowym,
- DBY B1,B2,...B30** — wyszukaj sekwencję bajtów B1,B2,...B30
- DHC tr,se,inc** — rozpocznij wyszukiwanie ciągłe,
- DHS tr,se,inc** — rozpocznij wyszukiwanie i zakończ je po pierwszym odnalezieniu zadanej sekwencji,
- DCC** — kontynuuj przerwane wyszukiwanie,
- DIN** — inicjalizacja początkowa w/w/wył.

Zwróć uwagę, że nie ma odpowiednika polecenia PDS w tym trybie pracy.

Parametry są następujące:

- tr** — nr ścieżki, na której znajduje się początek programu lub numer ścieżki, od której zaczynamy wyszukiwanie.
- se** — jak wyżej, lecz odnosi się do sektora.
- inc** — powiększony o 1 numer ścieżki, na której chcemy zakończyć wyszukiwanie.

Ponadto dostępne są następujące polecenia:

- DIR** — wyświetla katalog dyskietki, podając jednocześnie tr i se w trybie wyszukiwania w programie (wcisnij CTRL, aby kontynuować).
- DST** — wyświetla komunikat o statusie stacji (np. 00,OK,00,00)
- DCM ”polecenie”** — umożliwia wykonanie podanego polecenia DOS.

Podczas pracy stacja wyświetla następujące komunikaty:

- 00,FOUND BYTES,tr,se**
Sekwencja występuje
- 00,FOUND BYTES,tr,se,ltr,lse**
Sekwencja występuje w dwóch różnych sektorach (tzn. fragment sekwencji jest zapisany w jednym, a reszta w drugim sektorze).
- 00,NO MORE BYTES,tr,se**
Przeszukiwanie zakończone, sekwencja nie występuje.
- 24,READ ERROR,tr,se**
Błąd odczytu danego sektora powtórzył ponad 255 razy.

W komunikatach parametry tr i se oznaczają odpowiednio ścieżki i sektory, w których znajduje się poszukiwana sekwencja bądź zakończono jej wyszukiwanie; ltr i lse informują nas o numerze ścieżki i sektora, w których występuje drugi fragment sekwencji.

Jeśli poszukiwana sekwencja bajtów czy znaków występuje kilkakrotnie w przeszukiwanym sektorze, to nie jest to sygnalizowane. Stwarza to hipotetyczną możliwość pominięcia takiej sekwencji, gdy jest ona zawarta częściowo w dwóch kolejnych sektorach. W takim wypadku wystarczy jedynie sprawdzić ostatnie bajty podejrzanego sektora, korzystając z polecenia PHS. Następnie wykonujemy PDS i wiemy, czy nasze obawy były uzasadnione.

Program wykorzystuje w całości pamięć RAM stacji 1541 (jej bardzo uproszczoną mapę można znaleźć w instrukcji obsługi). „Szperacz” umieszcza w buforach #0, #1 i #2 procedurę wyszukiwania; pozostałe (#3 i #4) są wykorzystane do przechowywania zawartości badanych sektorów. Zawartość aktualnie przeszukiwanego sektora jest zapisana w buforze #3. Bufor #4 używany jest w trybie wyszukiwania sekwencji w programie. Sterowanie buforami odbywa się za pośrednictwem komórek: \$03 — wczytanie/sprowadzenie, \$0C — parametr tr, \$0D — parametr se dla bufora #3. W buforze #4 komórki te mają odpowiednio adresy \$04, \$0E i \$0F. Algorytm procedury wyszukującej jest w zasadzie zgodny z instrukcją H dowolnego monitora; jest on rozszerzony o możliwość wyszukiwania fragmentarycznej sekwencji występującej w dwóch sektorach.

Jeżeli dyskietka została wyposażona w „nielegalne” ścieżki (od 36 do 40), to prowadzenie na nich wyszukiwania jest możliwe. Pamiętaj jednak, że badane za pomocą szperacza oprogramowanie NIE MOŻE być zabezpieczone za pomocą celowo wprowadzonych błędów na dyskietce.

Czytelników zainteresowanych szczegółami lub rozszerzeniem możliwości programu zapraszamy do listownej wymiany informacji. Teraz pozostaje już tylko wykorzystać program do wprowadzania różnych usprawnień w grach.

*Piotr Barylak
Andrzej Urbankowski*

Od redakcji:

Niecierpliwym zalecamy, aby NIGDY nie „szperali” na dyskietce oryginalnej. Najpierw skopiujcie program i badania przeprowadzajcie wyłącznie na kopii. W ten sposób unikniecie utraty programu, w razie gdyby próby z jakichś przyczyn się nie powiodły.


```

10 REM *****
20 REM *
30 REM *      DISK SEARCHER V1.1
40 REM *
50 REM *      AUTOR: PIOTR BARYLAK
60 REM *
70 REM *      PRZY UDZIALE: URAN
80 REM *
90 REM *      (C) BAJTEK 1991
92 REM *
94 REM *****
100 :
110 :
120 :
130 :
140 :
150 PRINTCHR*(147)CHR*(17)CHR*(29)"  OBECNIE TRWA PRZE
PIBYWANIE DANYCH:PRINT
160 PRINTCHR*(29)"  URUCHOMIENIE PROGRAMU ZA 1 MIN."
170 AP=49152:AK=53231:FOR M=AP TO AK STEP 10
180 B=0:W=0
190 FOR N=0 TO 9
200 READ A:POKE M+N,A
210 B=B+A
220 NEXT N
230 READA:IF$(>ATHENM=1:PRINT:PRINT
240 IFW=1:THENPRINT"      ":CHR*(18)"POPRAW DANE W LINII
NR:";300*M-AP:STOP
250 NEXT M
260 SYS 49152:NEW
270 :
280 :
290 :
300 DATA 169,196,141,008,003,169,192,141,009,003, 1031
310 DATA 169,000,141,032,208,141,033,208,160,000, 1092
320 DATA 185,032,192,240,006,032,210,255,200,208, 1560
330 DATA 245,096,147,153,013,213,192,192,192,192, 1635
340 DATA 192,192,192,192,192,192,192,192,192,192, 1920
350 DATA 192,192,192,192,192,201,213,192,192,192, 1950
360 DATA 192,192,192,192,192,192,192,192,192,192, 1920
370 DATA 192,192,192,201,013,221,032,068,073,083, 1267
380 DATA 075,032,083,069,065,082,067,072,069,082, 696
390 DATA 032,086,049,046,049,221,032,032,032,067, 835
400 DATA 079,068,069,068,032,073,078,032,049,057, 605
410 DATA 056,057,032,221,013,171,192,192,192,192, 1318
420 DATA 192,192,192,192,192,192,192,192,192,192, 1920
430 DATA 192,192,192,192,192,192,192,192,192,192, 1886
440 DATA 192,192,192,192,192,192,192,192,192,192, 1920
450 DATA 192,192,192,203,013,202,192,192,192,192, 1762
460 DATA 156,032,065,067,084,073,086,065,084,069, 781
470 DATA 068,153,032,192,192,192,192,203,221,032, 1477
480 DATA 069,084,069,082,032,083,079,070,084,087, 739
490 DATA 065,082,069,013,158,000,032,115,000,240, 774
500 DATA 004,201,091,240,003,076,231,167,160,000, 1173
510 DATA 032,115,000,201,080,240,008,200,201,068, 1145
520 DATA 240,003,076,008,175,140,091,193,032,115, 1703
530 DATA 000,141,092,193,032,115,000,141,093,193, 1000
540 DATA 160,000,140,094,193,162,000,173,094,193, 1209
550 DATA 010,024,168,185,095,193,221,092,193,208, 1389
560 DATA 009,200,232,224,002,208,242,076,037,193, 1423
570 DATA 185,095,193,201,255,208,003,076,084,193, 1493
580 DATA 238,094,193,173,094,193,201,012,144,211, 1553
590 DATA 076,084,193,032,115,000,172,094,193,173, 1132
600 DATA 091,193,240,018,185,140,193,141,061,193, 1455
610 DATA 185,151,193,141,062,193,032,017,017,076, 1067
620 DATA 087,193,185,118,193,141,079,193,185,129, 1503
630 DATA 193,141,080,193,032,017,017,076,087,193, 1029
640 DATA 076,008,175,076,174,167,000,000,000,000, 676
650 DATA 000,072,067,072,083,067,067,083,067,065, 643
660 DATA 083,066,089,068,083,083,084,067,077,073, 773
670 DATA 082,073,078,255,193,196,199,202,205,208, 1691
680 DATA 211,181,184,187,214,193,193,193,193,193, 1942
690 DATA 193,193,193,193,193,193,162,165,168,171, 1824
700 DATA 174,177,180,181,184,187,190,193,193,193, 1852
710 DATA 193,193,193,193,193,193,193,193,193,077, 1697
720 DATA 194,076,080,194,076,111,194,076,039,196, 1236
730 DATA 076,033,196,076,130,195,096,076,226,199, 1303
740 DATA 076,050,202,076,045,197,076,217,193,076, 1208
750 DATA 014,195,076,017,195,076,042,195,076,027, 913
760 DATA 196,076,021,196,076,124,195,076,217,194, 1371
770 DATA 076,238,193,032,003,194,173,061,204,073, 1247
780 DATA 141,141,061,204,201,032,208,003,076,017, 1084
790 DATA 194,076,031,194,032,003,194,173,150,206, 1257
800 DATA 073,141,141,150,206,201,032,208,003,076, 1231
810 DATA 017,194,076,031,194,160,000,185,045,194, 1096
820 DATA 240,000,032,210,255,200,208,245,096,160, 1652
830 DATA 000,185,066,194,240,006,032,210,255,200, 1388
840 DATA 208,245,096,160,000,185,071,194,240,006, 1405
850 DATA 032,210,255,200,208,245,096,069,078,084, 1477
860 DATA 069,082,032,073,078,073,084,073,065,076, 705
870 DATA 073,090,065,084,073,079,078,000,032,079, 653
880 DATA 078,013,000,032,079,070,070,013,000,169, 524
890 DATA 001,044,169,000,141,252,202,032,136,195, 1172
900 DATA 169,000,141,250,202,173,199,195,141,244, 1714
910 DATA 202,173,200,195,141,245,202,173,201,195, 1927
920 DATA 141,246,202,173,250,202,240,003,076,013, 1546
930 DATA 195,032,254,202,032,226,199,192,003,208, 1543
940 DATA 015,173,045,202,141,244,202,173,046,202, 1443
950 DATA 141,245,202,076,172,194,173,048,202,205, 1658
960 DATA 245,202,208,008,173,047,202,205,244,202, 1736
970 DATA 240,025,173,047,202,141,244,202,173,048, 1495
980 DATA 202,141,245,202,173,252,202,240,087,192, 1936
990 DATA 001,240,194,192,003,240,005,169,001,141, 1186
1000 DATA 250,202,096,032,078,069,088,084,032,084, 1015
1010 DATA 082,065,067,075,032,035,000,032,044,032, 464
1020 DATA 083,069,067,084,079,082,032,035,000,160, 691
1030 DATA 000,185,191,194,240,006,032,210,255,200, 1513
1040 DATA 208,245,169,000,174,133,206,032,205,189, 1561
1050 DATA 160,000,185,205,194,240,006,032,210,255, 1487
1060 DATA 200,208,245,169,000,174,138,206,032,205, 1577
1070 DATA 189,169,013,076,210,255,192,001,208,169, 1482
1080 DATA 096,096,169,001,044,169,000,141,253,202, 1171
1090 DATA 032,136,195,169,000,141,251,202,173,199, 1478
1100 DATA 195,141,133,206,173,200,195,141,138,206, 1928
1110 DATA 173,251,202,208,222,032,036,203,032,226, 1585
1120 DATA 199,192,003,208,015,173,045,202,141,133, 1311
1130 DATA 206,173,046,202,141,138,206,076,100,195, 1483
1140 DATA 173,048,202,205,138,206,208,008,173,047, 1408
1150 DATA 202,205,133,206,240,025,173,047,202,141, 1574
1160 DATA 133,206,173,048,202,141,138,206,173,253, 1673
1170 DATA 202,240,014,192,001,240,194,192,003,240, 1518
1180 DATA 005,169,001,141,251,202,096,192,001,208, 1266
1190 DATA 242,096,032,136,195,076,045,196,032,136, 1186
1200 DATA 195,076,063,196,160,000,140,094,193,032, 1149
1210 DATA 180,195,076,153,195,032,115,000,032,183, 1161
1220 DATA 195,172,094,193,153,199,195,238,094,193, 1726
1230 DATA 224,000,208,005,173,094,193,208,232,172, 1509
1240 DATA 094,193,140,198,195,076,115,000,032,121, 1164
1250 DATA 000,032,158,183,168,138,192,093,240,003, 1207
1260 DATA 162,000,096,162,001,096,000,000,000,000, 517
1270 DATA 000,000,000,000,000,000,000,000,000,000, 0
1280 DATA 000,000,000,000,000,000,000,000,000,000, 0
1290 DATA 000,000,000,000,000,000,000,000,000,000, 0
1300 DATA 000,000,000,000,000,000,000,000,000,000, 0
1310 DATA 000,000,000,000,000,000,000,000,000,000, 0
1320 DATA 000,000,165,122,141,018,196,165,123,141, 1071
1330 DATA 019,196,096,238,018,196,208,003,238,019, 1231
1340 DATA 196,173,017,017,096,032,081,196,076,045, 929
1350 DATA 196,032,135,196,076,045,196,032,081,196, 1185
1360 DATA 076,065,196,032,135,196,076,063,196,160, 1193
1370 DATA 000,185,199,195,153,164,202,200,204,198, 1700
1380 DATA 195,208,244,140,163,202,096,160,000,185, 1593
1390 DATA 199,195,153,083,202,200,204,198,195,208, 1837
1400 DATA 244,140,082,202,096,032,254,195,032,017, 1294
1410 DATA 196,201,034,208,041,160,000,032,009,196, 1077
1420 DATA 201,034,240,064,153,199,195,200,208,243, 1679
1430 DATA 173,018,196,133,122,173,019,196,133,123, 1286
1440 DATA 140,198,195,032,115,000,201,093,240,003, 1217

```

```

1450 DATA 076,008,175,076,115,000,076,008,175,032, 741
1460 DATA 081,196,162,000,189,000,004,157,221,196, 1206
1470 DATA 169,032,157,000,004,232,224,080,208,240, 1346
1480 DATA 056,032,240,255,138,072,152,072,162,000, 1179
1490 DATA 160,000,024,032,240,255,160,000,185,199, 1255
1500 DATA 195,032,210,255,200,204,198,195,208,244, 1941
1510 DATA 160,000,185,000,004,153,199,195,200,204, 1300
1520 DATA 198,195,208,244,162,000,189,221,196,157, 1770
1530 DATA 000,004,232,224,080,208,245,104,168,104, 1369
1540 DATA 170,024,076,240,255,234,234,234,234,234, 1935
1550 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1560 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1570 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1580 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1590 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1600 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1610 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1620 DATA 234,234,234,234,234,169,001,162,196,160, 1858
1630 DATA 197,032,189,255,169,008,170,168,032,186, 1406
1640 DATA 255,032,192,255,032,183,255,201,128,240, 1773
1650 DATA 100,169,008,170,168,032,198,255,160,000, 1260
1660 DATA 140,195,197,140,194,197,032,207,255,200, 1757
1670 DATA 192,254,208,248,160,000,032,207,255,153, 1709
1680 DATA 197,197,032,183,255,208,064,200,192,030, 1558
1690 DATA 208,240,238,194,197,160,000,185,197,197, 1816
1700 DATA 200,201,000,240,248,192,002,176,042,173, 1474
1710 DATA 194,197,201,008,240,027,032,207,255,032, 1393
1720 DATA 207,255,032,002,198,173,001,220,201,255, 1544
1730 DATA 240,249,032,225,255,240,014,238,195,197, 1885
1740 DATA 076,094,197,169,000,141,194,197,076,142, 1286
1750 DATA 197,160,000,185,090,199,240,006,032,210, 1319
1760 DATA 255,200,208,245,032,204,255,169,008,170, 1746
1770 DATA 032,195,255,096,000,000,036,000,080,069, 763
1780 DATA 084,069,082,032,083,079,070,084,087,065, 735
1790 DATA 082,069,032,069,078,084,069,082,234,234, 1033
1800 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1810 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1820 DATA 234,234,234,234,234,234,234,234,234,234, 2340
1830 DATA 234,234,234,234,234,234,234,234,173,195, 2240
1840 DATA 197,208,015,160,000,185,145,198,240,008, 1356
1850 DATA 032,210,255,200,192,255,208,243,169,032, 1796
1860 DATA 160,000,153,131,199,200,192,040,208,248, 1531
1870 DATA 160,000,169,221,153,131,199,160,038,153, 1384
1880 DATA 131,199,160,001,162,003,189,197,197,153, 1392
1890 DATA 131,199,200,232,192,017,208,244,160,019, 1602
1900 DATA 162,001,032,181,199,160,233,162,002,032, 954
1910 DATA 181,199,160,032,162,029,032,181,199,200, 1375
1920 DATA 162,028,032,187,199,173,197,197,041,127, 1343
1930 DATA 010,010,168,162,027,185,125,198,157,131, 1173
1940 DATA 199,200,232,224,031,208,244,160,000,185, 1683
1950 DATA 131,199,032,210,255,200,192,040,208,245, 1712
1960 DATA 096,068,069,076,032,083,069,081,032,080, 686
1970 DATA 082,071,032,085,083,082,032,082,069,076, 694
1980 DATA 032,213,192,192,192,192,192,192,192,192, 1781
1990 DATA 192,192,192,192,192,192,192,192,192,192, 1920
2000 DATA 192,192,192,192,192,192,192,192,192,192, 1920
2010 DATA 192,192,192,192,192,192,192,192,192,201, 1929
2020 DATA 013,221,069,088,084,069,078,068,069,068, 827
2030 DATA 032,068,073,082,069,067,084,079,082,089, 725
2040 DATA 032,058,032,032,080,069,084,069,082,032, 570
2050 DATA 083,079,070,084,032,049,057,056,057,221, 788
2060 DATA 013,171,192,192,192,192,192,192,192,192, 1720
2070 DATA 192,192,192,192,192,192,192,192,192,178, 1906
2080 DATA 192,192,192,178,192,192,192,192,178,192, 1892
2090 DATA 192,192,178,192,192,192,192,192,192,179, 1893
2100 DATA 013,221,032,078,065,077,069,032,058,032, 677
2110 DATA 032,032,032,032,032,032,032,032,032,221, 509
2120 DATA 084,082,075,221,083,069,067,221,084,089, 1075
2130 DATA 080,069,221,066,076,079,067,075,083,221, 1037
2140 DATA 013,171,192,192,192,192,192,192,192,192, 1720
2150 DATA 192,192,192,192,192,192,192,192,192,177, 1905
2160 DATA 192,192,192,177,192,192,192,192,192,192, 1890
2170 DATA 192,192,177,192,192,192,192,192,192,179, 1892
2180 DATA 013,000,202,192,192,192,192,192,192,192, 1559
2190 DATA 192,192,192,192,192,192,192,192,192,192, 1920
2200 DATA 192,192,192,192,192,192,192,192,192,192, 1920
2210 DATA 192,192,192,192,192,192,192,192,192,192, 1920
2220 DATA 203,013,000,234,234,234,234,234,234,234, 1854
2230 DATA 234,234,234,234,234,234,234,234,234,234, 2340
2240 DATA 234,234,234,234,234,234,234,234,234,234, 2340
2250 DATA 234,234,234,234,234,234,234,234,234,234, 2340
2260 DATA 234,234,234,234,234,234,234,234,234,234, 2340
2270 DATA 234,234,234,169,036,153,131,199,200,189, 1779
2280 DATA 197,197,072,041,240,032,073,199,104,041, 1326
2290 DATA 015,200,076,208,199,074,204,074,074,024, 1018
2300 DATA 201,010,176,007,024,105,048,153,131,199, 1054
2310 DATA 096,024,105,055,153,131,199,096,169,003, 1031
2320 DATA 141,091,193,169,000,032,189,255,169,015, 1254
2330 DATA 168,162,008,032,186,255,032,192,255,169, 1459
2340 DATA 015,170,032,198,255,160,000,032,207,255, 1324
2350 DATA 153,091,201,032,183,255,208,003,200,208, 1534
2360 DATA 242,140,214,201,174,091,193,224,003,240, 1722
2370 DATA 003,076,167,200,162,000,160,003,185,091, 1047
2380 DATA 201,201,044,200,008,232,224,003,208,003, 1332
2390 DATA 076,051,200,200,204,214,201,208,253,140, 1729
2400 DATA 090,201,160,001,185,091,201,201,045,208, 1383
2410 DATA 090,162,000,160,000,185,091,201,201,044, 1134
2420 DATA 208,005,232,224,003,240,009,153,141,201, 1416
2430 DATA 200,204,214,201,208,235,162,000,153,141, 1718
2440 DATA 201,200,140,191,201,162,000,172,214,201, 1682
2450 DATA 185,091,201,136,201,044,208,248,232,224, 1770
2460 DATA 002,208,243,174,191,201,200,200,185,091, 1695
2470 DATA 201,157,141,201,232,200,204,214,201,208, 1959
2480 DATA 243,142,191,201,160,000,185,141,201,032, 1496
2490 DATA 210,255,200,204,191,201,208,244,076,167, 1956
2500 DATA 200,160,000,185,091,201,032,210,255,200, 1534
2510 DATA 204,090,201,208,244,169,013,032,210,255, 1626
2520 DATA 032,204,255,169,015,032,195,255,032,197, 1386
2530 DATA 200,140,049,202,192,004,176,006,032,009, 1010
2540 DATA 201,172,049,202,096,160,002,162,000,189, 1233
2550 DATA 192,201,240,050,217,091,201,208,006,200, 1606
2560 DATA 232,224,255,208,240,160,002,162,000,189, 1672
2570 DATA 204,201,240,033,217,091,201,208,004,200, 1599
2580 DATA 232,208,242,160,003,162,000,189,209,201, 1606
2590 DATA 240,018,217,091,201,208,004,200,232,208, 1619
2600 DATA 242,160,000,096,160,001,096,160,002,096, 1013
2610 DATA 160,003,096,16
```


pakiet procedur w Turbo Pascalu

Pakiety pascalowych procedur graficznych dostępne są dziś do prawie każdego kompilatora na prawie każdym komputerze. Również Spectrum, dzięki stacji dysków FDD 3000 może używać Turbo Pascala, a więc i wielu procedur pascalowych niedostępnych „zwyktemu” Spectrum.

Rzadko zdarza się, by pakiet graficzny miał możliwość rysowania obrazów trójwymiarowych a tym bardziej, by była to prawdziwa perspektywa, a nie rzut równoległy.

Prezentuję bibliotekę procedur, która wykorzystując tylko dwie dostępne szeroko procedury **line (x1,y1,x2,y2)** i **plot (x1,y1)**, pozwala uzyskać trójwymiarowe, perspektywiczne obrazy. Program ten oferuje jednak dość skromne możliwości — rysowanie punktów, linii oraz prostopadłościanów o krawędziach równoległych do osi układu współrzędnych. Dlatego proponuję rozszerzenie pakietu o możliwości takie, jak operowanie płaszczyznami, co pozwoli na dalsze udoskonalanie programu np. o likwidację niewidocznych linii, czy cieniowanie.

LISTING 1 Biblioteka POWER.SYS

```
function power(podst,wykl:real):real;
{FUNKCJA OBLICZA POTEGE O PODSTAWIE}
{I WYKŁADNIKU RZECZYWISTYM. UZYWA JEJ}
{PAKIET GRAPH-3D.SYS. FUNKCJA JEST PO-}
{ŁACZENIEM FUNKCJI POT NAPISANEJ PRZEZ}
{P. JANA BOBROWSKIEGO Z KRAKOWA ("KOM-}
{PUTER" 6/1989, STR. 30) I PROCEDURY}
{DOPOT AUTORSTWA P. LECHA SZCZEPANIAKA}
{Z BIAŁEGO STOKU ("KOMPUTER" 6/1988,}
{STR. 25)}

var calk:integer;
    ulam,y1,y2:real;

function intpower(p:real;w:integer)
:real;
var y:real;
begin
    y:=1;
    while w<>0 do
    begin
        if (w and 1)=1 then y:=y*p;
        w:=w shr 1;
        if w<>0 then p:=sqr(p)
    end;
    intpower:=y;
end;

begin
    calk:=round(int(wykl));
    ulam:=wykl-calk;
    y1:=intpower(podst,calk);
    y2:=exp(ulam*ln(podst));
    power:=y1*y2
end;
```

LISTING 2 Biblioteka GRAPH-3D.SYS

```
var sxg,syg,szg:integer;
{WSPÓLRZĘDNE KURSORA GRAFICZNEGO}
{W PRZESTRZENI}
sxgl,sygl:integer;
{J.W., ALE NA PŁASZCZYZNIE}
axisx,axisy:integer;
{WSPÓLRZĘDNE PUNKTU, W KTORYM ZBIE-}
{GAJA SIĘ WSZYSTKIE LINIE W SPÓL-}
{RZĘDNYCH EKRANOWYCH}

procedure axis(x,y:byte);
```

Osie układu współrzędnych usytuowane są następująco: **Y** jest osią pionową, **X** poziomą, a **Z** biegnie w głąb ekranu.

Listing 1 (biblioteka **POWER.SYS**) zawiera funkcję podnoszenia liczby rzeczywistej do potęgi rzeczywistej. Jest ona połączeniem procedury **DOPOT** („Komputer” 6/88, str. 25) i funkcji **POT** („Komputer” 6/89, str. 30).

Listing 2 (**GRAPH-3D.SYS**) to właściwy pakiet procedur. Aby użyć go w swoim programie, na początku należy napisać:

```
{$i graph.sys} — dołączenie pakietu procedur gra-
ficznych (Bajtek 11 12/90)
{$i power.sys}
{$i graph-3d.sys}
```

Podstawowe zmienne użyte w pakiecie, to:

sxg, syg, szg — współrzędne ostatnio narysowanego punktu (XYZ),

sxgl, sygl — j.w., w ekranowym układzie współrzędnych,

axisx, axisy — współrzędne ekranowe punktu, w którym zbiegają się wszystkie linie równoległe do osi Z

Procedury i funkcje:

xe(x,z : integer) : integer — funkcja przelicza współrzędne przestrzenne x i z na współrzędną ekranową x.

ye(y, z : integer) : integer — j.w., ale y i z przelicza na y.

```
{USTAWIENIE PUNKTU, W KTORYM ZBIEGAJA}
{SIE LINIE}
begin
    axisx:=x;
    axisy:=y
end;

function xe(x,z:integer):integer;
{FUNKCJA PRZELICZA WSPÓLRZĘDNE PRZE-}
{STRZENNE NA WSPÓLRZĘDNA EKRANOWA X PRZY}
{POMOCY WZOROW NA JEDNOKŁADNOŚĆ}
begin
    xe:=round(power(0.4,z*0.005)*
        (x-axisx)+axisx)
end;

function ye(y,z:integer):integer;
{J.W., ALE NA WSPÓLRZĘDNA Y}
begin
    ye:=round(power(0.4,z*0.005)*
        (y-axisy)+axisy)
end;
```

```
procedure splot(x,y,z:integer);
{RYSOWANIE PUNKTU W PRZESTRZENI, PRZE-}
{SUNIECIE KURSORA GRAFICZNEGO}
begin
    sxg:=x;syg:=y;szg:=z;
    sxgl:=xe(x,z);sygl:=ye(y,z);
    plot(sxgl,sygl,1)
end;
```

```
procedure sdraw(dx,dy,dz:integer);
{RYSOWANIE ODCINKA W PRZESTRZENI OD}
{PUNKTU (SXG,SYG, SZG) DO PUNKTU}
{(SXG+DX,SYG+DY, SZG+DZ), PRZESUNIECIE}
{KURSORA GRAFICZNEGO}
begin
    sxg:=sxg+dx;syg:=syg+dy;szg:=szg+dz;
    line(sxgl,sygl,xe(sxg,szg),
        ye(syg,szg),1);
    sxgl:=xe(sxg,szg);sygl:=ye(syg,szg)
end;
```

```
procedure sdrawto(x,y,z:integer);
{J.W., ALE DO PUNKTU (X,Y,Z)}
begin
    sdraw(x-sxg,y-syg,z-szg)
end;

procedure sline(x1,y1,z1,
                x2,y2,z2:integer);
{J.W., ALE OD PUNKTU (X1,Y1,Z1) DO PUN-}
{KTU (X2,Y2,Z2)}
begin
    sxg:=x2;syg:=y2;szg:=z2;
```

splot(x,y,z : integer) : integer — rysowanie punktu o zadanych współrzędnych w trójwymiarowym układzie, przesunięcie kursora

sdraw(dx,sy,dz : integer) : integer — rysowanie odcinka od miejsca, w którym znajduje się kursor graficzny do miejsca odległego o współrzędne wektora

sdrawto(x,y,z : integer) — j.w., ale do punktu o danych współrzędnych

sline(x1,y1,z1,x2,y2,z2 : integer) — rysowanie odcinka od punktu (x1,y1,z1) do (x2,y2,z2)

sbox(x1,y1,z1,x2,y2,z2) : integer — rysowanie prostopadłościanu o wyznaczonej punktami przekątnej

Uwaga! Aby dopasować powyższy pakiet do swojego komputera i do używanego pakietu procedur graficznych, wystarczy zmienić nazwy i ew. parametry procedur **plot** i **line**. Użytkownicy wyższych wersji Turbo Pascala i Quick Pascala mogą bez trudu napisać odpowiedni moduł (unit).

Przedstawiony na listingu 3 plik **DEMO-3D.PAS** demonstruje użycie pakietu, rysując trzy proste rysunki. Jeżeli program ma być uruchomiony np. na IBM PC, na początku trzeba dopisać komendę włączającą tryb graficzny. Na Spectrum jest to zbędne.

Marcin Sawicki
(lat 13)

```
sxgl:=xe(x2,z2);sygl:=ye(y2,z2);
line(xe(x1,z1),ye(y1,z1),sxgl,sygl,1)
end;

procedure sbox(x1,y1,z1,
               x2,y2,z2:integer);
{RYSOWANIE PROSTOPADŁOŚCIANU O DANYCH}
{WSPÓLRZĘDNYCH PRZEKĄTNEJ}
begin
    sline(x1,y1,z1,x2,y1,z1);
    sdrawto(x2,y2,z1);
    sdrawto(x1,y2,z1);
    sdrawto(x1,y1,z1);
    sdrawto(x1,y1,z2);
    sdrawto(x2,y1,z2);
    sdrawto(x2,y2,z2);
    sdrawto(x1,y2,z2);
    sdrawto(x1,y1,z2);
    sline(x1,y2,z1,x1,y2,z2);
    sline(x2,y2,z1,x2,y2,z2);
    sline(x2,y1,z1,x2,y1,z2)
end;
```

LISTING 3 Plik DEMO-3D.PAS

```
program demo3d;
{PROGRAM DEMONSTRUJE UŻYCIE PAKIETU}
{GRAPH-3D.SYS RYSUJĄC 3 PROSTE TRÓJ-}
{WYMIAROWE RYSUNKI}

{$i graph.sys}
{$i power.sys}
{$i graph-3d.sys}

var a,b:integer;

begin
    clrscr;
    axis(60,100);
    sbox(0,0,0,255,175,175);
    for a:=1 to 10 do
        sline(0,0,a*16,255,0,a*16);
    for b:=1 to 15 do
        sline(b*16,0,0,b*16,0,175);
    repeat until keypressed;
    clrscr;
    axis(128,88);
    for a:=0 to 2 do
        sbox(40,0,a*200,215,175,a*200+175);
    repeat until keypressed;
    clrscr;
    axis(0,88);
    for a:=0 to 2 do
        sbox(40,0,a*200,215,175,a*200+175);
    repeat until keypressed
end.
```


Smok i geś

czyli jeszcze raz

TOS I GENS 3M21

Jeśli masz program GENS 3M21 to w przeciągu godzinny możesz stać się posiadaczem jego nowej wersji — GENS 3M3 — w której wszystkie rozkazy współpracy z magnetofonem zostały opcjonalnie poszerzone o możliwość pracy z dyskiem. GENS 3M3 ma taką samą długość jak jego pierwowzór, czyli 10034 bajty i jest oczywiście nadal programem relokowalnym.

W Bajtku był już opisywany problem współpracy asemblera GENS 3M21 z TOS-em. Zamieszczone procedury miały jedną podstawową zaletę — że działały. Były one bardzo potrzebne i dzięki nim ten doskonały asembler mógł być dużo efektywniej wykorzystywany. Jednak z czasem pojawiają się problemy. Dajmy na to pracujemy nad jakimś większym programem, którego część ma być w BASIC'u, a część w kodzie maszynowym. Okazuje się, że nie możemy pisać programu, ponieważ system BASIC'a jest zajęty przez podprogramy obsługi dysku dla GENS'a! Z tym problemem jednak jakoś sobie poradzimy. Jednak gdy mamy w komputerze 5 KB wolnej pamięci, a do skompilowania 10 KB kodu źródłowego, to najlepszy nawet program BASIC'owy nie umożliwi

nowy plik o nazwie GENS 3M3. Jes to już gotowa, końcowa wersja programu.

Oto kilka uwag, które mogą być przydatne:

1. Próby przerabiania innych wersji, np. GENS 3 lub GENS 3M2, są z góry skazane na niepowodzenie.

2. Jeśli Twój GENS 3M21 ma z jakichś powodów zmienioną nazwę, na przykład rozszerzenie .COD, to należy oczywiście odpowiednio zmodyfikować 10 linię programu.

Obsługa edytora nowo powstałego asemblera nie różni się właściwie niczym od swego poprzednika. Tak samo się uruchamia, nawet ilość poleceń się nie zmieniła. Do współpracy ze stacją dysków używamy tych samych komend jak przy magnetofonie. Wyróżnikiem jest gwiazdka stawiana przed nazwą bloku poddawanego operacjom we/wy.

KOMENDY EDYTORA:

G „*nazwa — wczytuje (lub doczytuje do istniejącego) tekst

G „* — wyświetla katalog dyskietki

P n1, n2,*nazwa — zapisuje kod źródłowy od linii n1 do n2

O,*nazwa — zapisuje kod wynikowy KOMENDY ASEMBLERA:

***F *nazwa** — kompilacja z dysku UWAGI:

1. W przypadku współpracy z dyskiem nie ma podziału — jak w przypadku taśmy — na teksty przeznaczone dla edytora, oraz teksty w formacie specjalnie zmienionym do kompilacji. Za pomocą komendy ***F** asemblera możemy kompilować te same pliki, które wysłaliśmy komendą **P**. Z tego też powodu nie istnieje opcja komendy **T** dla stacji dysków.

2. Procedura kompilacji z dysku korzysta z bufora o stałej długości 256 bajtów, dlatego też przy ewentualnej zmianie długości buforów przy pytaniu „Buffer size? wystarczy wcisnąć ENTER.

3. W przypadku błędu obsługi stacji GENS 3M3 informuje o tym przy pomocy standardowych komunikatów TOS-u.

4. GENS 3M3 nie posiada możliwości zmiany napędu, ani też poruszania się po podkatalogach. W celu naprawy tej wady podczas wykonywania komendy **G„*** (bez nazwy) GENS sprawdza, czy w pamięci znajduje się program **CAT** (patrz BAJTEK 11—12/90). Jeśli tak — to uruchamia go, jeśli zaś nie — to wypisuje standardowy katalog dyskietki.

```
10 LOAD *"GENS3M21"CODE 32768
20 LET suma=0: FOR A=1 TO 454: READ B: LET suma=suma+b: IF B>255 THEN LET AD=B: NEXT A
30 POKE AD,B: LET AD=AD+1: NEXT A: IF suma<>745651 THEN PRINT "Bład w liniach DATA.": STOP
```

```
40 SAVE *"GENS3M3"CODE 32768,10034
50 STOP
100 DATA 34687,177,30,253,229,253
110 DATA 33,0,0,207,253,225,197,213,17,0
120 DATA 32,126,237,160,254,13,32,249,175,27
130 DATA 18,19,123,209,193,201,62,18,205,72
140 DATA 11,205,163,8,58,4,33,50,46,33
150 DATA 17,1,1,205,66,11,192,17,5,0
160 DATA 205,148,13,192,126,254,3,194,168,8
170 DATA 34900,24,47
180 DATA 34952,42,32,62,229,193,205,126,7
190 DATA 8,175,50,52,33,62,3,205,97,10
200 DATA 205,163,8,205,49,0,195,4,6,58
210 DATA 2,33,167,200,253,33,58,92,33,13
220 DATA 33,205,216,3,205,70,11,205,4,6,195,47,0
230 DATA 35074,176,30
240 DATA 201,42,22,31,62,128,188,62,255,210
250 DATA 113,9,253,229,253,33,0,0,207,253
260 DATA 33,58,92,33,160,34,221,229,209,213
270 DATA 193,183,237,82,229,17,162,33,237,82
280 DATA 227,121,183,195,253,9
290 DATA 35325,40,4,54,1,237,176,98,107,209
300 DATA 229,205,148,13,209,253,225,32,29,237
```

```
310 DATA 176,98,107,43,1,128,0,62,13,237
320 DATA 185,62,127,145,79,197,221,225,35,35
330 DATA 54,0,35,54,0,195,4,6,254,72
340 DATA 194,163,8,120,177,32,218,225,225,33
350 DATA 18,24,227,205,70,11,24,233
360 DATA 35405,58,176,30,254,42,32,63
370 DATA 205,126,7,254,1,32,22,58,104,33
380 DATA 254,6,202,104,33,62,11,205,37,7
390 DATA 205,136,6,205,49,0,195,4,6,71
400 DATA 42,54,0,34,80,33,33,0,0,34
410 DATA 78,33,62,3,50,77,33,205,193,12
420 DATA 42,71,33,205,163,8,245,205,4,6,24,37
```

```
430 DATA 36838,254,42,32
440 DATA 49,253,203,2,198,253,126,1,50,33
450 DATA 31,253,203,1,214,235,35,205,129,7
460 DATA 205,149,3,253,229,253,33,58,92,205
470 DATA 158,7,253,225,194,163,8,205,4,6
480 DATA 33,255,255,221,33,0,0,195,47,16
490 DATA 38806,5
500 DATA 40510,77,41126,51
510 DATA 41139,40,87,74,41
520 DATA 41645,62,30,78,10,2,9,85,10
530 DATA 142,8,41687,115,10,138,10
540 DATA 41711,155,8,12,16
550 DATA 164,7,47,10,150,23,7
560 DATA 16,242,15,251,15
570 DATA 41765,25,16,190,7,185,8,127,7,6,9,149,28,9,40,9,48,9,56,10
580 DATA 160,9,160,9,160,9,160,9,160,9,160,9
590 DATA 42165,160,9,160,9,160,9,160,9
600 DATA 42523,160,9
610 DATA 42737,160,9
```

Wielu z Was zapewne zastanawia się, jak do tak zwartego programu jak GENS 3M21 "wcisnąć" kilkaset bajtów dodatkowego kodu? Przecież to niemożliwe! Zgadza się. Trzeba było coś usunąć. Otóż program ten — o czym z zasady nie pamięta się — posiada możliwość współpracy z microdrive'm i właśnie w miejsce procedur jego obsługi zostały wstawione podprogramy obsługi stacji dysków.

Oto kilka rad, które uchronią Cię przed zbędnym zdenerwowaniem podczas obsługi asemblera:

1. Przy wysyłaniu kodu źródłowego

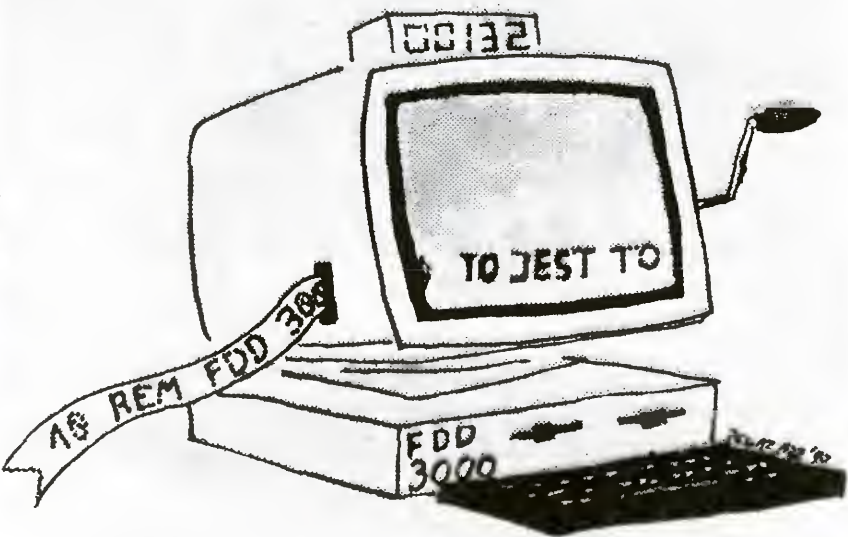
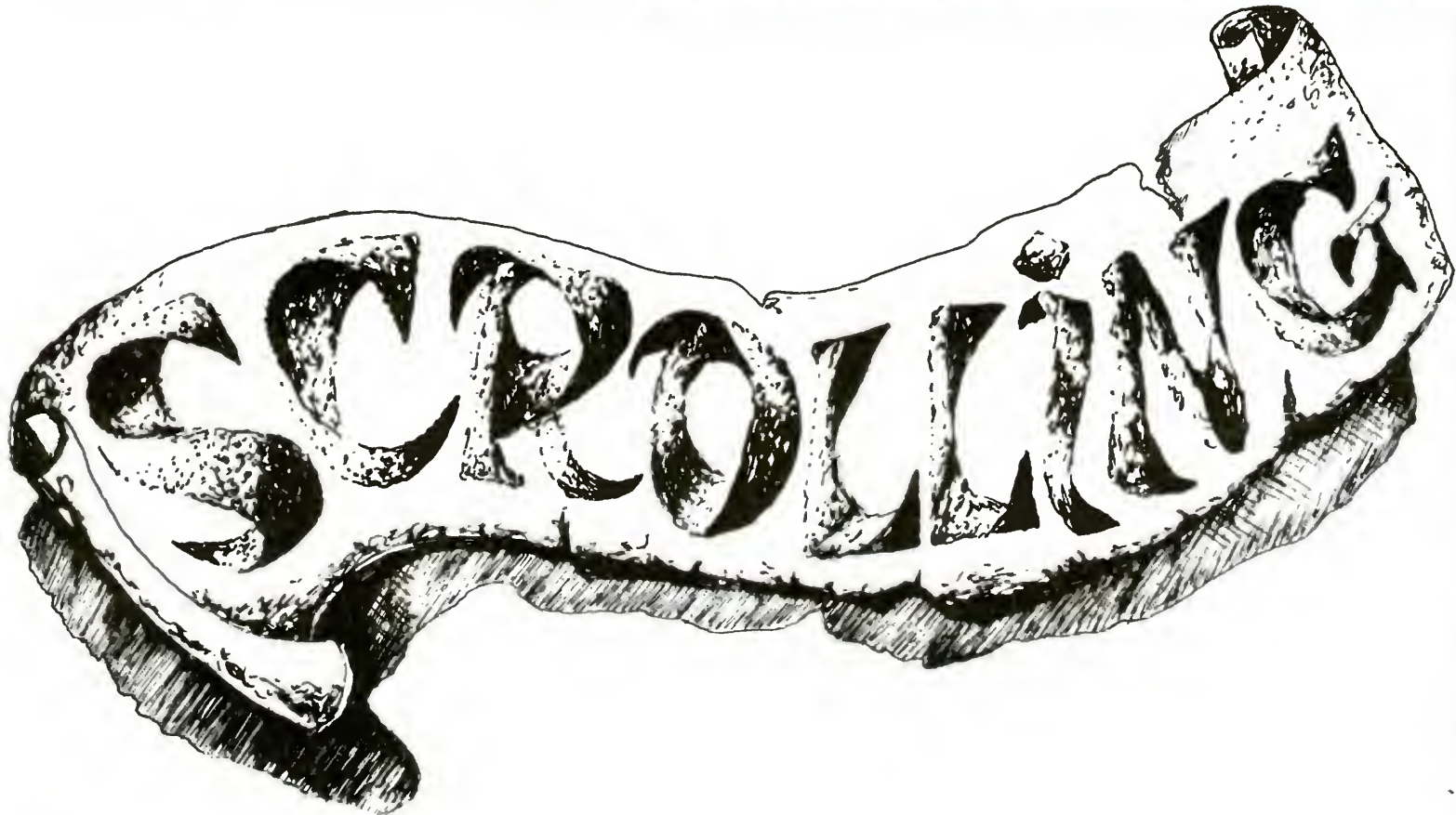
(komenda **P**) zawsze podawaj numery linii; zapobiegnie to ewentualnej stracie części tekstu.

2. Jeśli chcesz wgrać do edytora nowy tekst, musisz najpierw skasować stary (za pomocą komendy **D n1, n2**).

3. Przy łączeniu tekstów z dysku GENS 3M3 nie renumeruje linii; musisz zrobić to sam.

Jeśli weźmiesz pod uwagę powyższe rady i przymrużysz oko na trochę śmieszny edytor, typowy dla produktów firmy HiSoft, to otrzymasz naprawdę dobry w pracy asembler.

Wojciech Jabłoński



Każdy program powinien przedstawić swojego autora. W jaki sposób będzie to zrobione, zawsze i wyłącznie zależy od programującego. Dobrze, gdy autor może zaprezentować swój oryginalny pomysł. Lecz gdy go nie posiada, pozostaje skorzystanie z gotowego przepisu.

Najprostszą formą przekazania kilku słów we własnym programie jest wydrukowanie ich w postaci tzw. **scrollingu** — ruchomych napisów poruszających się na ekranie z prawej strony do lewej. Nie jest to reguła — tekst może poruszać się zupełnie dowolnie, byle był możliwy do odczytania. W ten sposób można przekazać tekst liczący wiele zdań, który nie zmieściłby się w całości jednocześnie na ekranie. Ze scrollingu korzysta wiele gier, a także prawie wszystkie programy — dema wykorzystujące generator dźwięku AY.

My zaczniemy od najprostszych metod poruszania napisów, przy okazji poznając reguły ich tworzenia.

Już na początku można zauważyć, że Basic mimo, że potrafi drukować, to do szybkiego przesuwania napisów się nie nadaje. Trzeba wybrać inny język. Najlepszym będzie assembler, a to dlatego, że: jest na pewno wystarczająco szybki i można w nim napisać program tak, aby był on zsynchronizowany z tworzeniem się obrazu na ekranie telewizora — jest to potrzebne, aby przesuwany napis nie drgał, tylko płynnie się poruszał.

Następnie trzeba się zdecydować jaką wielkość mają mieć wyświetlane litery, skąd będziemy pobierać ich wzorce (z ROM-u, czy zdefiniujemy własny zestaw znaków o wymiarach innych niż 8*8 punktów), jaki będzie kierunek przesuwania znaków (poziomy, pionowy, sinusoidalny), czy wygląd znaków ma się zmieniać w trakcie przesuwania (pochylanie, obracanie wokół własnej osi) i na koniec metodę przesuwania. Powyższe parametry mają wpływ na komplikację programu i jego długość.

Dla prostoty wybierzmy oryginalny zestaw znaków z ROM-u ZX Spectrum i kierunek przesuwania poziomy z prawej strony do lewej. Aby program nie był zbyt prosty, spróbujmy powiększyć drukowane znaki dwu- lub czterokrotnie.

Można wybrać jedną z dwóch metod drukowania: Pierwsza polega na drukowaniu od razu całego znaku na prawym krańcu ekranu, ale znak trzeba przykryć atrybutami tak, aby w obszarze znaku kolor atramentu był równy kolorowi papieru. Następnie przesuwamy żądany fragment ekranu w prawo tyle razy, aby cały znak wysunął się spod przykrywających go atrybutów. Teraz w tym samym miejscu co poprzedni znak drukujemy następny i tak w pętli, aż do końca tekstu.

Druga metoda wymaga połączenia procedury przesuwającej z drukującą tak, aby w momencie przesunięcia istniejącego tekstu w lewo wydrukować następny pionowy pasek (o szerokości jednego punktu) nowego znaku. Trzeba jeszcze zapamiętać, jaka część znaku została już wydrukowana.

Trudniejszą metodą jest druga. Spróbujmy ją przeanalizować:

Przesuwanie zaczynamy od prawego górnego rogu znaku. Najpierw trzeba wyliczyć, czy punkt, który pojawi się na ekranie będzie zapalony, czy zgaszony. Następnie trzeba go wydrukować znajdując adres pamięci ekranu, w którym zostanie on umieszczony. Pod tym adresem przesuwamy każdy **bit** o jedną pozycję w lewo, jako **bit 0** wstawiamy nowy punkt, a zapamiętujemy wysunięty **bit nr 7**. Teraz przesuwamy się o jeden adres w pamięci ekranu w lewo i powtarzamy przesuwanie bitów, z tą różnicą, że na **bit nr 0** wpisujemy zapamiętany, stary, **bit nr 7**.

Ten fragment programu można łatwo napisać posługując się rozkazami przesunięć bitowych przez akumulator i znacznik przeniesienia (CARRY). Wykonujemy w pętli 32 takie przesunięcia, aż dotrzemy do lewego brzegu ekranu. Następnie trzeba wrócić na prawy brzeg ekranu, ale o jedną linię niżej i wykonać jeszcze raz te same czynności. Po

wykonaniu w pętli tylu przesunięć linii ekranu jaka jest wysokość znaków mamy na ekranie tekst przesunięty o jeden punkt w lewo i wystarczy jeszcze jedna duża pętla dla kolejnych pikseli dla następnych znaków, aby zaczął się on płynnie przesuwać.

W tym momencie okazuje się, że komputer nic innego nie robi tylko przesuwa tekst. Coś trzeba zmienić. Na przykład można zamiast pętli wywoływać przesuwanie ekranu na przemian z wykonywanym głównym programem. Pojawia się problem odmierzania stałych przedziałów czasu na oba programy, bo inaczej tekst nie będzie się przesuwał płynnie, tylko skokowo. Najlepiej przesuwanie tekstu zlecić przerwaniu maskowalnemu, które co 1/50 sekundy przerwie główny program i przesunie tekst o jeden punkt. Ten pomysł ma jeszcze jedną zaletę, a mianowicie przerwanie maskowalne pojawia się równocześnie z początkiem nowego obrazu na ekranie, zatem ruch tekstu będzie zsynchronizowany z tworzeniem obrazu. W ten sposób można uniknąć nieprzyjemnego drgania tekstu.

W praktyce mogłoby się okazać, że tekst przesuwa się wolniej niż 50 punktów na sekundę, bo program przesuwający trwa tyle czasu, że gubi następne przerwania. Wtedy trzeba odwrócić metodę, to znaczy główny program wykonywać na przerwaniach (o ile to możliwe), a przesuwanie tekstu poza nimi. Tą metodą posługują się dema na generator dźwięku AY — muzyka odtwarzana jest na przerwaniach, a tekst poza przerwaniami.

Jeśli chcemy szybciej przesuwać tekst można obliczać i przesuwać w lewo od razu dwa, trzy kolejne bity.

Na podstawie powyższych zaleceń powstał program przedstawiony na **listingu 1**. Osoby, które nie posiadają żadnego assemblera mogą wpisać program w postaci linii danych, z **listingu 2**, ale wówczas nie będą mogły go przystosować do własnych potrzeb.

Program nie jest relokowalny, ale wpisanie postaci źródłowej do assemblera pozwala zmienić adres kompilacji. Sam program działa dokładnie według powyższego opisu i warto go prześledzić ze względu na zastosowany algorytm. Niektóre fragmenty mogą przydać się w innych programach, np. warto zapamiętać metodę instalowania przerwania maskowalnego wykorzystującego nieużywany fragment pamięci ROM na tablicę wektorów przerwań (linie 240—410) oraz procedurę wyliczania adresu linii o punkt niżej w pamięci ekranu ZX Spectrum (linie 600—710).

Główną część programu stanowi procedura **NEXT**, która oblicza, czy wsuwane od prawej strony piksele mają być zapalone, czy zgaszone. Każde jej wywołanie zwraca w znaczniku przeniesienia CARRY informację o jednym bicie kolejnego znaku.

Program wykorzystuje komórki 65524 — 65535 jako swoje zmienne. Zmienne te są ustawiane w liniach 190—220. Zmienna **T1** wskazuje na początek tekstu i jest ustawiona wstępnie na adres komentarza w pierwszej linii Basica:

1 REM to jest tekst, który chcę przesuwać «ENTER»

Można to zmienić zmieniając stałą **TEKST**, albo linie 190—220. Zmienna **T2** wskazuje na aktualnie drukowaną literę tekstu, a zmienna **Q** na fragment litery. Zmienna **POS** wskazuje na prawy górny brzeg przesuwanego okna tekstu i jest ustawiona na adres pierwszej linii znaku o współrzędnych AT 20,31 w pamięci ekranu. Można zmienić tę współzrzedną zmieniając stałą **POSIT**, ale musi to być adres komórki pamięci z prawego brzegu ekranu.

Tekst jest drukowany od początku po napotkaniu znaku CHR\$ 13 czyli kodu ENTER, który kończy linię w Basicu (patrz linie 780—840). Przykładowy program drukujący tekst jest przedstawiony na **listingu 3**. Komenda **USR 65000** uruchamia przesuwanie tekstu na przerwaniach, a **USR 65002** wyłącza przesuwanie. Program drukuje poprawnie tylko znaki z ROM-u (CHR\$ 32 do CHR\$ 127 włącznie) powiększając każdy znak dwukrotnie. Zmieniając stałą **FONT** można używać swojego zestawu znaków, a wtedy można używać wszystkich kodów.

Można również drukować znaki powiększone czterokrotnie, ale tekst będzie się bardzo wolno przesuwał. W tym celu należy zmienić wartości stałych w liniach: 480, 750, 910 i dodać dwie linie 750 i 1080. W komentarzach podane są konieczne zmiany.

Prezentowany program powinien być częścią większego programu np. dema i to niekoniecznie napisanego w assemblerze.

LISTING 1

```
10 TEKST EQU 23760 ;Adr. te
<stu
20 POSIT EQU 20639 ;Adr. prawego gorn
ego rogu tekstu
30 FONT EQU 15360 ; Adr. zestawu zna
kow
40
42 ;Adresy zmiennych programu
50 T1 EQU #FFF7
60 POS EQU #FFF9
70 T2 EQU #FFFB
80 Q EQU #FFFD
90
95 ;Początek programu
100 ORG 65000
110 START JR START1
120
125 ;Wyłączenie przerwan
130 STOP DI
140 IM 1
150 EI
160 RET
170
175 ;Inicjalizacja przerwan i zmiennych
180 START1 DI
190 LD HL,TEKST
200 LD (T1),HL
210 LD HL,POSIT
220 LD (POS),HL
230
240 LD A,#39
250 LD I,A
260 LD HL,#FFFF
270 LD (HL),#18
280 LD HL,#FFF4
290 LD (HL),#C3
300 LD DE,SCROLL
310 INC HL
320 LD (HL),E
330 INC HL
340 LD (HL),D
350 LD HL,(T1)
360 LD (T2),HL
370 LD HL,Q
380 LD (HL),0
390 IM 2
400 EI
410 RET
420
425 ;Początek obsługi przerwania
430 SCROLL PUSH AF
440 PUSH HL
450 PUSH BC
460 PUSH DE
470 LD HL,(POS) ;Adr. prawego b
rzedu napisu
475 ;Ilość linii napisu
480 LD B,16 ;32
485 ;Wwołanie procedury obliczającej wsuw
any na ekran bit
490 SC1 CALL NEXT
500 PUSH HL
510 PUSH BC
520 LD B,32
525 ;Przesunięcie jednej linijki w petli

530 LINE LD A,(HL)
540 RLA
550 LD (HL),A
560 DEC HL
570 DJNZ LINE
580 POP BC
590 POP HL
595 ;Obliczenie adresu niższej linii
600 INC H
610 LD A,H
620 AND 7
630 JR NZ,LOOP
640 LD A,L
650 ADD A,32
660 LD L,A
```

Marek Sawicki

JĘZYK MASZYNOWY

cz. 5

```

670      CCF
680      SBC  A,A
690      AND  248
700      ADD  A,H
710      LD   H,A
715      ;Powtarzaj az do ostatniej linii
720      LOOP DJNZ SC1
725      ;Przesuniecie o piksel w aktualnym z
naku
730      LD   A,(Q)
740      INC  A
750      AND  15      ;31
760      LD   (Q),A
770      JR   NZ,END
775      ;Przesuniecie sie na nastepny znak
780      LD   HL,(T2)
790      INC  HL
800      LD   A,13
805      ;Czy to koniec tekstu?
810      CP   (HL)
820      JR   NZ,N_CHAR
825      ;Zaczniij od poczatku tekstu
830      LD   HL,(T1)
840      N_CHAR LD (T2),HL
845      ;Powrot z przerwania poprzez test kl
awiatuury
850      END      POP  DE
860      POP  BC
870      JP   #003A
880
885      ;Procedura obliczajaca wsuwany z pra
wej
886      ;strony bit i zwracajaca go w CARRY
890      NEXT      PUSH HL
900      PUSH BC
910      LD   A,16      ;32
915      ;Numer piksela w znaku liczac od gor
y
920      SUB  B
930      ;      SRL  A
940      SRL  A
950      LD   HL,(T2)
955      ;Kod ASCII znaku
960      LD   L,(HL)
970      LD   H,0
980      ADD  HL,HL
990      ADD  HL,HL
1000      ADD  HL,HL
1010      LD   DE,Font
1015      ;Adr. matrycy znaku
1020      ADD  HL,DE
1030      LD   D,0
1040      LD   E,A
1045      ;Adr. linii w znaku
1050      ADD  HL,DE
1060      LD   A,(Q)
1070      LD   B,A
1080      ;      SRL  B
1090      SRL  B
1100      INC  B
1110      LD   A,(HL)
1115      ;Bit w linii znaku
1120      SHIFT SLA A
1130      DJNZ SHIFT
1140      POP  BC
1150      POP  HL
1160      RET
1170      ;W CY zwracany jest wyliczony bit

```

LISTING 2

```

1 DATA 65000,6
2 DATA "1805F3ED58FBC9F321D05C22F7FF219
F5022F9FF3E39ED4721FFFF361821F4FFC5"
3 DATA "38C31120FE237323722AF7FF22FBFF2
1FDF3600ED5EFBC9F5E5C5D52AF9FF068D"
4 DATA "10CD69FEE5C506207E17772B10FAC1E
1247CE607200A7DC6206F3F9FE6F8846727"
5 DATA "10DF3AFDFF3CE60F32FDF200F2AFBF
F233E0DBE20032AF7FF22FBFFD1C1C33AF1"
6 DATA "00E5C53E1090CB3F2AFBFF6E2600292
92911003C1916005F193AFDFF47CB38043D"
7 DATA "7ECB2710FCC1E1C932021B"
1000 CLEAR 64999: PRINT "OK": PAUSE VAL "1
00": BEEP SGN PI,SGN PI: CLS : PRINT "
WAIT A MOMENT!"
1010 GO SUB 9992
1020 SAVE "SCROLL.1"CODE 65000,172
1030 STOP
9992 RESTORE
9993 READ A,S: FOR F=SGN PI TO S
9994 READ L*: LET L=LEN L*: LET S=NOT PI:
LET K=VAL "2"
9995 LET A*=L*(K-SGN PI): LET B*=L*(K)
9996 LET C=(CODE A*-VAL "48"-(VAL "7"* (A*
"0")))*VAL "16"+CODE B*-VAL "48"-(VAL "7"*
(B* "0"))
9997 IF K<L THEN POKE A,C: LET S=S+C: LET
K=K+VAL "2": LET A=A+SGN PI: GO TO VAL "9
995"
9998 IF S-VAL "256"*INT (S/VAL "256")<>C T
HEN PRINT "BLAD W LINII ";F+SGN PI: STOP
9999 NEXT F: RETURN

```

LISTING 3

```

1 REM Pozdrowienia dla czytelnikow "BAJTKA".
10 CLEAR 64999
20 LOAD "SCROLL.1"CODE 65000
30 PAPER 0: INK 7: BORDER 1: CLS
40 PRINT AT 10,8;"Wcisnij klawisz"
50 RANDOMIZE USR VAL "65000"
60 PAUSE 0
70 RANDOMIZE USR VAL "65002"

```

Po przybliżeniu spraw potrzebnych do zrozumienia działania procedury NEXT-ONE z ROM-u Spectrum, przejdziemy do jej omówienia. Procedura rozpoczyna się od adresu 6584 (hex. 19B8).

Celem procedury jest, na podstawie zadanego adresu wejściowego (umieszczanego w HL), określenie, czy pod tym adresem znajduje się linia programu, czy zmienna oraz jaka jest jej długość. Automatycznie otrzyma się adres następnej linii programu lub zmiennej.

Parametrem wejściowym jest umieszczany w HL adres, a parametry wyjściowe to: długość linii/zmiennej w BC oraz adres następnej linii/zmiennej w DE.

Działanie procedury:

1. Rozpoznanie linia/zmienna.

Wykorzystywany jest fakt, że pierwszy bajt linii jest mniejszy od 64, pierwszy zaś bajt każdej ze zmiennych większy od 64. Następuje porównanie zawartości akumulatora pobranej spod adresu w HL z 64. Ustawienie znacznika przeniesienia CY powoduje skok dalej (rozkaz 4), przeciwnie realizowany jest program dla zmiennych.

2. Podział na zmienne 2, 4, 6 oraz 3, 5, 7

Jeśli procedura „obrabia” zmienną, rozkazy 5 i 6 dokonują jej rozdziału na grupę 2, 4, 6 lub 3, 5, 7. Podstawą rozdziału jest fakt, że pierwsza grupa posiada wyzerowany 5-ty bit, druga zaś zapalony. Skok (rozkaz 6) wykonuje się dla grupy 2, 4, 6, a nie wykonuje się dla grupy 3, 5, 7.

3. Rozdział na zmienne 5 oraz 3, 7

Zmienne typu nieparzystego nie posiadają zapisanej długości, stąd dalsze postępowanie jest różne dla każdej z nich. Rozkazy 7 i 8 dokonują rozdziału na typy 5 i 3, 7 wykorzystując fakt, że pierwszy bajt zmiennej 5 zawiera się w przedziale (161, 186), dzięki czemu po podwojeniu tych wartości (rozkaz 7) nastąpi przepelnienie i ustawiony zostanie znacznik CY, lecz nadmiar znajdujący się w akumulatorze posiada wartość od 66 do 116, czyli jest zawsze mniejsza od 128. Dzięki temu nie jest ustawiany wskaźnik znaku (P), co rozpoznaje rejestr wskaźników F, więc skok (rozkaz 8) nie wykonuje się. Ten szczegół jest charakterystyczny tylko dla zmiennej typu 5, co radzę sprawdzić.

4. Rozdział na zmienne 3, 5 oraz 7

Rozdział ten realizuje rozkaz 11 korzystając z tego, że po ADD A,A poza zapaleniem siódmego bitu znaku dla zmiennej 3 i 7 nie jest ustawiony znacznik przepelnienia dla zmiennej typu 3. Znacznik CY jest też zmieniany (rozkaz 9), aby w tym miejscu przebieg programu był właściwy. Dzięki temu następuje jednoznaczny podział na zmienne 3, 5 oraz 7.

Wcześniej wykonywany jest rozkaz 10 (konieczny dla zmiennej typu 3, a nie przeszkadzający zmiennym typu 5 i 7).

5. Przygotowanie parametrów dla obliczenia długości linii/zmiennej

Blok ten rozciąga się od rozkazu 7 do 17 i dotyczy wyłącznie zmiennych typu 3, 5 i 7. Dla zmiennej 3 realizowane są rozkazy 7, 8, 10, 11, jednorazowo 13, 14, 15 i 17.

Dla zmiennej typu 5 realizowane są rozkazy 7, 9, 10, 11, k-krotnie rozkazy 13—16 oraz rozkaz 17. Omijany jest rozkaz 8.

Dla zmiennej typu 7 realizowane są rozkazy 7, 8, 10, 11, 12, jednorazowo 13—15 oraz rozkaz 17.

6. Ustalenie długości i adresu następnej danej

Etap ten realizowany jest podobnie dla wszystkich siedmiu rodzajów danych, lecz każda wymaga osobnego przygotowania pewnych parametrów.

Po rozpoznaniu linii działanie przenoszone jest do rozkazu 18, gdzie HL zwiększane jest o 2, by otrzymać adres wskazujący młodszy bajt długości linii (ładowany do rejestru C). Bajt starszy umieszczany jest w B. Liczba ta (BC) oznacza jednak długość linii bez czterech pierwszych bajtów. W celu otrzymania adresu następnej linii, rozkaz 23 ustala w HL adres piątego bajtu linii i dodaje doń wartość BC. Z kolei, po odjęciu od obliczonego adresu następnej linii adresu wejściowego, otrzymuje się pełną długość linii, która umieszczana jest w BC. Proponuję dokładne przeanalizowanie powyższego fragmentu procedury.

W przypadku zmiennych typu 2, 4 i 6 ustalenie adresu następnej zmiennej przebiega niemal identycznie jak dla

linii. Jedyna różnica to skok wykonywany tu rozkazem 6, przenoszący działanie do rozkazu 18, gdyż długość zmiennych typu 2, 4 i 6 zapisana jest w drugim i trzecim bajcie.

Dla zmiennej 5 nie wykonuje się rozkaz 8, dzięki czemu wykonywany jest rozkaz 11. Postępowanie polega na polczeniu znaków tworzących zmienną. Realizuje to pętla, przebiegająca od rozkazu 13 (pomnożenie akumulatora przez 2) do 16 (skok, gdy nie ma przeniesienia). Ustawienie znacznika CY następuje, gdy w akumulatorze pojawi się kod ostatniego znaku oznaczenia zmiennej.

W przypadku zmiennej typu 3 problem nie istnieje, gdyż posiada ona stałą długość równą 6. Jednak procedura jest na tyle uniwersalna, że normalną drogą potrafi przeanalizować i tę zmienną. Pierwszy bajt po podwojeniu (rozkaz 7) przekracza 127, lecz nie przekracza 255, stąd znacznik znaku S=1, a przepelnienia CY=0. Dzięki temu omijane są rozkazy 14 i 15, do BC ładowana liczba 5. Rozkaz 16 nie ma żadnego znaczenia, ale jest realizowany z konieczności.

Ostatni typ zmiennej 7 posiada również stałą długość, tu wynoszącą 19 bajtów. Pierwszy bajt zmiennej może przyjmować wartości od 225 do 250, więc po podwojeniu otrzymujemy zawsze ustawiony znacznik przepelnienia oraz bit znaku. Podobnie jak dla zmiennej 3, gdzie do BC ładowana była liczba 5, tak tu do BC wchodzi 18, a nie 19. W obu przypadkach przyczyną jest „czekający” rozkaz 14, który zwiększa liczbę w BC o 1.

Uwagi:

Omawiana procedura NEXT-ONE, choć z pozoru mało przydatna, służyć może do wielu celów. Natychmiast nasuwają się:

- policzenie linii programu,
- sprawdzenie, czy istnieje w programie linia o danym numerze,
- sprawdzenie, czy istnieje linia o danej długości,
- znalezienie adresu określonej linii lub zmiennej,
- sprawdzenie istnienia zmiennej,
- policzenie zadeklarowanych zmiennych.

Dokładna analiza procedury potwierdza, że „sposób myślenia” procesora jest daleki od sposobu myślenia człowieka, wymaga bowiem wielu subtelnych manipulacji i karkołomnych chwytów. Czy procedura NEXT-ONE to jedyny sposób rozwiązania problemu? Nie, można to zrobić inaczej, lecz czy inne rozwiązanie zajmie mniej niż 45 bajtów?

Proponuję przeanalizowanie równie krótkiej procedury, umieszczonej pod adresem 6510 (hex. 196E) o nazwie LINE-ADDR, będącą procedurą wyszukującą adres linii programu w Basic-u, której numer podano w HL. Korzysta ona z procedury NEXT-ONE i jest to jeden z dziewięciu takich przypadków w ROM-ie Spectrum.

Piotr Sumara

```

1 NEXT-ONE      PUSH HL
2              LD   A,(HL)
3              CP   64
4              JR   C,NEXT-0-3
5              BIT  5,A
6              JR   Z,NEXT-0-4
7              ADD  A,A
8              JP   M,NEXT-0-1
9              CCF
10 NEXT-0-1      LD   BC,0005
11              JR   NC,NEXT-02
12              LD   C,18
13 NEXT-0-2      RLA
14              INC  HL
15              LD   A,(HL)
16              JR   NC,NEXT-0-2
17              JR   NEXT-0-5
18 NEXT-0-3      INC  HL
19 NEXT-0-4      INC  HL
20              LD   C,(HL)
21              INC  HL
22              LD   B,(HL)
23              INC  HL
24 NEXT-0-5      ADD  HL,BC
25              POP  DE
26              AND  A
27              SBC  HL,DE
28              LD   B,H
29              LD   C,L
30              ADD  HL,DE
31              EX   DE,HL
32              RET

```


DIZZY — to najlepszy gracz w pokera w krainie EGGLAND. Kiedyś zagrał małą partyjkę z nowym gościem knajpy „Pod Gotowaną Kurą”. Z początku Dizziem nie szło za dobrze. Przegrał 2000 eggerolców, lecz w końcu uśmiechnął się do niego los. W jednym momencie odegrał wszystko i ograł gościa z ostatniej koszuli.

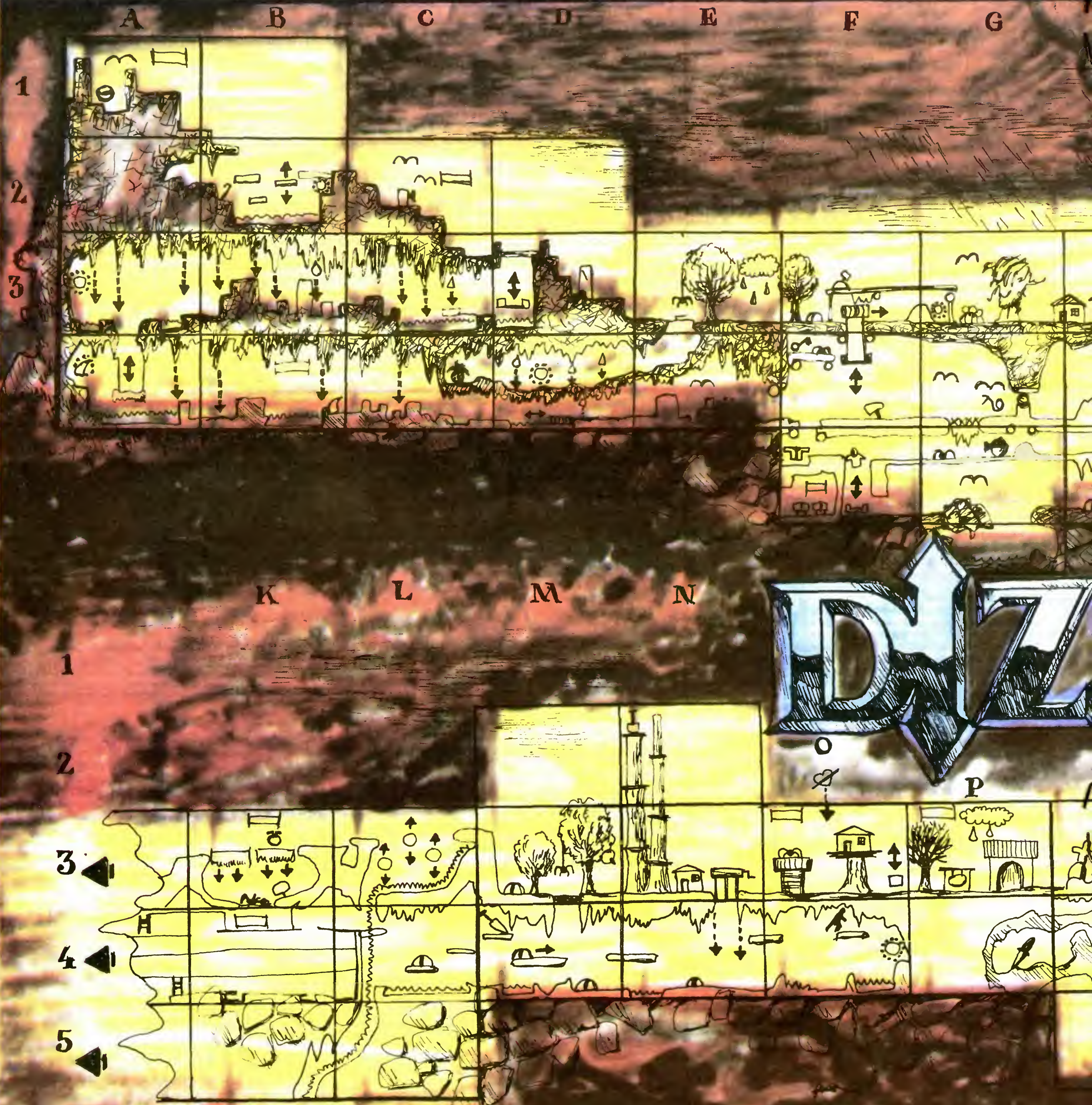
Kiedy zbierał się do domu, nieznajomy jakby od niechcenia ruszył małą pałeczką i... Dizzy obudził się w domu. Nie miał przy duszy ani grosza, a co gorsza — nie miał też ostatniej koszuli. W knajpie dowiedział się, że jego partner do gry zniknął.

Dalsze śledztwo wykazało, że owym graczem był najbardziej złośliwy czarnoksiężnik, jaki się

urodził we wszechświecie. Jego imię Painman było wymawiane ze strachem. Nikt nie wiedział, gdzie znajduje się jego siedziba. Dizzy postanowił jednak go odnaleźć. I tu zadanie dla Ciebie! Stajesz się najlepszym na świecie pokerzystą w Egglandzie — Dizzym.

Dizzy dowiedział się, że istnieje sposób na zlikwidowanie złego czarodzieja. Wystarczy odzyskać składniki najmocniejszej trucizny znanej na świecie. Wędrując po świecie, Dizzy znalazł kota, który świetnie się nadawał do przygotowania trucizny. Został mu problem, w czym tę truciznę przechowywać. Idąc w prawo znalazł butelkę. Wracając zauważył pod ścianą ledwo tłącą się pochodnię. Niestety, jego małe ręce pozwalają mu na noszenie tylko jednego przedmiotu. Szybciutko zaniósł butelkę do kotła i po-

wrócił po pochodnię. Gdy położył ją obok kociołka, z radością zauważył, że ten mały ogarek rozpałił płomień wystarczający do zagotowania wody. Rozpoczął poszukiwania składników diabelskiego eliksiru. W wiosce cygańskiej znalazł strzelbę przerobioną na oliwiarę. Idąc dalej w lewo, dotarł do starej kopalni złota. Wejście do niej zastawał stary wózek. Dizzy użył całej swojej siły, ale wózek nie drgnął. Ale od czego ma się oliwiarę? W podziemiach znalazł poza okropnymi nietoperzami klucz do furki zamkowej, laser na duchy i bardzo dobry płaszcz przeciwdeszczowy. W jednej z grot odnalazł nowiutką puszkę oliwy. Zwiedzając całą krainę, stanął przed wrotami zamku, lecz most zwodzony był podniesiony. I tu Dizzy znowu pomyślał — po co mu była puszką? Most z hukiem opadł. Siedziba Painmana była otwarta.



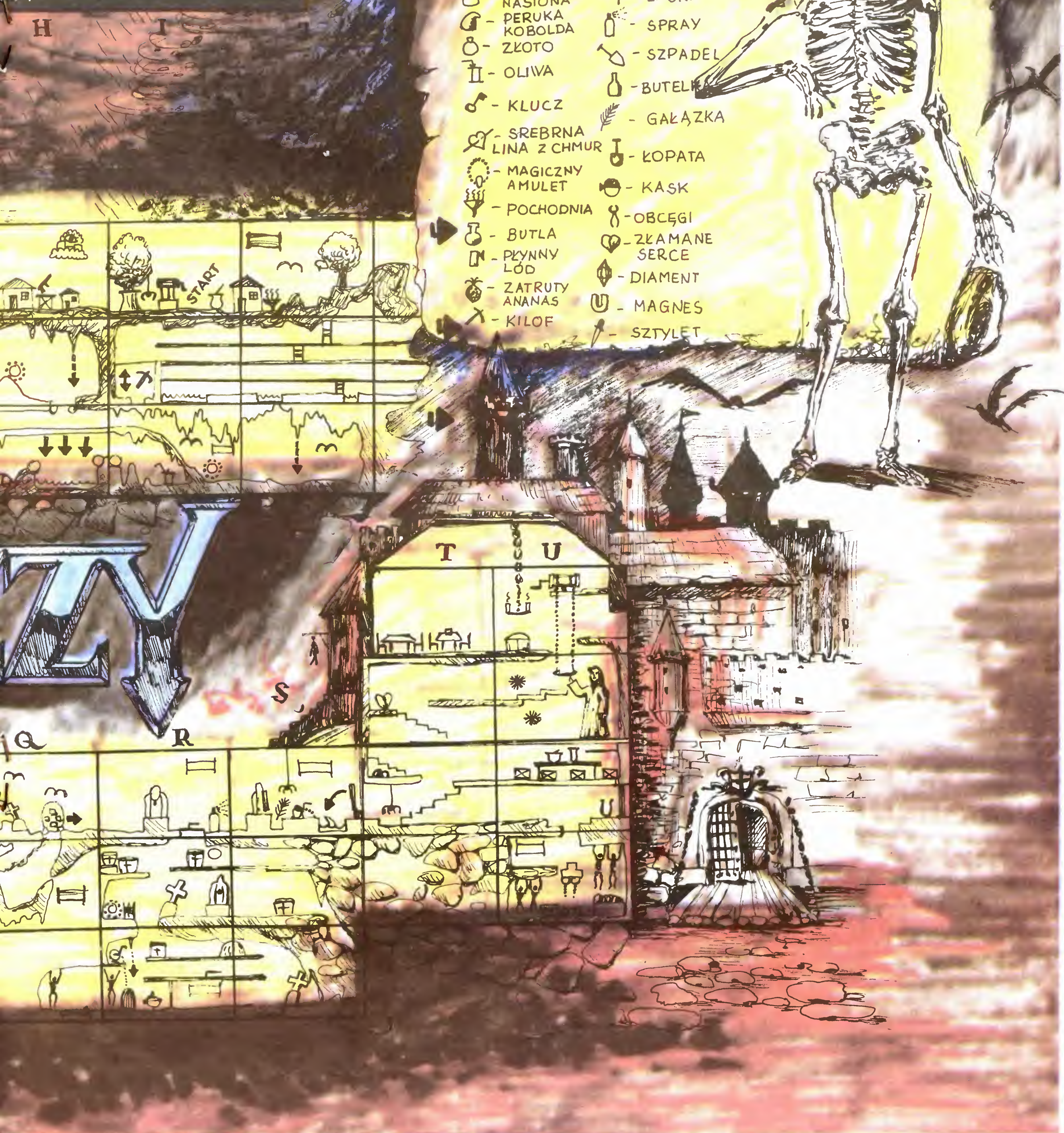
BECIA & PEGAZ ASS

Na stoku góry Dizy odnalazł żelazną łaskę. Jego umysł pracował na wysokich obrotach, wynikiem tego było otworzenie klapy przy drzewie. W podróży przeszkadzały mu okropne pająki, znalazł na nie sposób — zatrute nasiona ananasa, pająki — spray na pająki, spadające błąka i okruchy skalne — kask budowlany.

Nie psujmy jednak zabawy. Mapa z zaznaczonymi przedmiotami powinna pomóc w grze. Zawsza powinna być długa i miła, choć miejscami trochę denerwująca. Radzimy uważnie czytać wszystkie napisy umiejscowione w poszczególnych krainach. Do zobaczenia na Wyspie Skarłów i w Fantastycznym Świecie Dizziego!

Krzychu

Forma: CodeMasters
Komputer: Amstrad, Commodore, Spectrum
Scenariusz: Przemek i Sebastian



złota super

Oto obiecane zestawienie notowań Bajtkowej Listy Przebojów z lat 1986—1990. Pięć lat comiesięcznych testów daje bardzo dobry wgląd w Wasze upodobania co do gier komputerowych. Po świeże mięso zapraszamy jak zwykle do Top Secret!

1.

Operation Wolf
2.

Batman
3.

Boulder Dash
4.

Robocop
5.

Bomb Jack
6.

Platoon
7.

Knight Lore
8.

Spy vs Spy II
9.

Saboteur II
10.

Spy vs Spy
11.

Broadsides
12.

Untouchables

13.

Total Eclipse
14.

Solo Flight
15.

Wizard's Lair
16.

R-Type
17.

Winter Games
18.

Beach Head II
19.

Silent Service
20.

Arkanoid
21.

West Bank
22.

Sky Fox
23.

Trap Door
24.

Secret Diary of Adrian Mole
25.

Dan Dare
26.

Exploding Fist
27.

Exploding Fist III
28.

Iron Lord
29.

Hacker
30.

Jumping Jack

Za miesiąc ostatnie wydanie rubryki „Co Jest Grane” w starej szacie, a od numeru 4/91 wspaniała

niespodzianka!



Poszukuję gier TRAZ, BMX+, ACE OF ACES, BARBARIAN na Atari XE w wersji kasetowej. Posiadam około 250 gier (także w systemach Turbo 2000, AST, ATT). **Cezariusz Guth, ul. Młyńska 11a, 89-100 Nakło n/Notecią**
Pilnie poszukuję dokładnego opisu do gry CROMBLE'S CRISIS na Atari 130 XE. W zamian inne opisy. **Tomasz Worobiow, ul. Klasztorna, 36/59 33-300 Nowy Sącz**
Nie wiem, jak skończyć grę ROBIN HOOD, a także PIT-XFALL II i JAMES BOND na Atari 65 XE. Proszę o pomoc. **Sylwia Bartosiak, ul. Daleka 5/17, 15-037 Białystok, tel. 413-295**
Proszę o pomoc w grze NIGHT RAIDER — nie wiem, jak wystartować z lotniskowca. Komputer Spectrum. **Przemek Cyran, ul. Lilli Wenedy 11/73, 0-833 Kraków**
Jak w grze COLONY na Atari wymieniać części robota, aby przedłużyć jego żywotność? W zamian opis do FXEU-DA ze wskazówkami, jak w kilka minut wygrać tę grę. **Sławomir Chojnowski, ul. Żuławska 1c/7, 80-062 Gdańsk**
Mam Atari 800 XL. Szukam gier: BARBARIAN, MOLECULE MAN, TOMAHAWK BOUNTY BOB STRIKES BACK i MINER 2049 — wszystkie na dysk. W zamian inne. **Tomasz Depa, ul. Podwisłocze 18/163, 35-310 Rzeszów**
Poszukuję gier GHOSTS'N GOBLINS, POP EYE, HERBERT'S DUMMY RUN. W zamian inne gry i opisy. **Michał Jarosz, ul. Różyckiego 3c/9, 41-400 Mysłowice**
Założyliśmy własny mini-klub o nazwie „Club-Star Computers”. Mamy Commodore VC20, C+4 i Atari 800XL. Jesteśmy zainteresowani wymianą programów

(gier) i doświadczeń o każdym z tych komputerów. Prosimy o kontakt osobisty lub listowny. **Marcin Górecki, ul. Łukasińskiego 60, 43-300 Bielsko-Biała**
Mam komputer Atari XE. Potrzebuję opisu do planszy F w grze TWILIGHT WORLD, w wersji kasetowej. W zamian opisy do gier. **Daniel Szczechowski, ul. Borzymowska 41/7, 03-565 Warszawa.**
Pragnę wymienić się grami na komputer Atari 130 XE. W zamian oferuję m.in. Silent Service, Spy vs Spy I, II, III, Road Race, Beach Head I, II. **Tomasz Szedziński, ul. Kośnego 34a/8, 45-056 Opole**
Od dłuższego czasu szukam opisów do gier: SNOOKIE, TRAIL BLAZER, GATWAY TO APASTAI na Atari 800 XL. Za pomoc będę wdzięczny. **Artur Bednarczyk, ul. M. Spisaka 80, 02-495 Warszawa**
Mam 14 lat i komputer Timex 2048. Proszę o opis i dokładny plan do gry 4 x 4 OFF ROAD RACING. O co chodzi w grze MASK I—II? **Dawid Zimny, ul. Graniczna 110/3, 41-506 Chorzów**
Proszę o pomoc w grzec QUEST FOR TIRES i o dokładny opis gry SPELLBOUND. **Łukasz Mostowski, ul. Lompy 3/8, 44-100 Gliwice**
Poszukuję gier PLATOON, SUPER HUEY, KING'S QUEST, a także opisów do nich. **Marcin Zimny, ul. Wielkopolska 53/37, 44-335 Jastrzębie Zdrój**
Koledzy gracze, pomóżcie! Nie wiem, jaki kod wpisać w grze NIGHT GUNNER, aby ją uruchomić. Proszę również o opisy do gier UNDERWURDLE, MAG MAX oraz SHADOWFIRE. Posiadam Spectrum 48. **Marcin Pośpiech, ul. Stalowska 4 m 7, 53-404 Wrocław**
W zamian za gry: SUBMARINE COMMANDER, SUPER HUEY i THE TRAIN oferuję grę SILENT SERVICE wraz z opisem. **Wojciech Wolny, ul. Sprinterów 7 m 56, 94-002 Łódź**
Jestem posiadaczem komputera Atari 65XE. Nie wiem, jak wpisać się nieśmiertelności do gier. Proszę o pomoc. **Aleksander Martin, ul. Zamiejska 3 m 6, 03-580 Warszawa**
Pilnie poszukuję opisów, map i POKE'ów do następują-

cych gier: JET SET WILLY II, PANAMA JOE, DYNAMITE DAN na ZX Spectrum 48. W zamian inne. **Patryk Chycki, ul. Gałeczki 45 m 180, 41-500 Chorzów**
Poszukuję gier BARBARIAN, PLATOON, ELEVATOR ACTION, CONAN, THE TRAIN, IKARI WARRIORS i DRACONUS. W zamian 10 dowolnie wybranych ze 150. Komputer Atari 65XE. **Rafał Popek, ul. Zachodnia 53 m 5, 91-063 Łódź**
Od niedawna posiadam komputer Commodore 64. Nie wiem o co chodzi w grze CYBORG. Jak pokonać krokodyla w grze THE LAST NINJA? Jak przejść 8 etap w BOB'N RUMBLE. **Sebastian Wygas, ul. Warcella 32 m 11, 42-200 Częstochowa**
W zamian za grę THE GOONIES oddam wiele innych. **Jacek Śniadecki, ul. Bartoszewicza 1c m 8, 00-372 Warszawa**
Pilnie szukam opisów do gier CAVERNS OF KHAWKA, BROADSIDES, SPY vs SPY III. W zamian opisy do wielu innych gier. **Krzysztof Wychowalek, ul. Zbiorcza 6 m 18, 92-328 Łódź**
Mam Atari 65XE. Szukam opisów gier UNIVERSAL HERO, SPELLBOUND i MASTER OF THE LAMPS. Za pomoc dużo opisów do różnych gier. **Przemysław Kuźny, ul. Kosynierów Kościuszkowskich 1b m 26, 87-100 Toruń**
Jestem posiadaczem komputera Timex 2048. Poszukuję następujących gier: COMMANDO II, PREDATOR, INTERNATIONAL KARATE+, RAMBO III. Proszę także o przysłanie mi programów użytkowych THE ART STUDIO, TURBO PASCAL 3.0. **Krzysztof Kobus, ul. Chłopska 8, 85-355 Bydgoszcz**
Proszę o opisy gier F-15 STRIKE EAGLE, JUMBO JET PILOT, SUBMARINE COMMANDER i JUMP JET. **Piotr Endzel, ul. Sonaty 2 m 1010, 02-744 Warszawa**
Poszukuję gier BOB'N RUMBLE, ACE OF ACES, BARBARIAN i STREET FIGHTER na Spectrum +. Nie wiem też o co chodzi w grze LIVINGSTONE. Za gry i opis zapłacę gotówką. **Rafał Górski, ul. Wrocławska 199, 62-800 Kalisz**



SERVE & VOLLEY

Nie wydaje mi się przesadą pogląd, że dany kraj charakteryzuje częściowo to, jaki sport jest w nim najbardziej popularny. Brutalni Amerykańcy uwielbiają American Football, ludy wschodnie od zarania dziejów przywykły do popisów sztuk walki, Polacy widzą tylko, lub prawie tylko, piłkę nożną.

Patrząc na to obiektywnie, nic w tym dziwnego. Przyszyczenia są częścią ludzkiej natury — jak ktoś od małego ogląda w telewizji tylko piłkę nożną, to i tylko w nią w przyszłości grać będzie. Wpływają na to też troszeczkę warunki: brak dobrych sal gimnastycznych, kiepscy trenerzy, itp.

Dlatego też, między innymi, staramy się propagować takie gry sportowe, które wnoszą do naszych twardych głów

choć trochę nowego. Był więc Baseball, Koszykówka, Golf, a będzie jeszcze Siatkówka, Rugby, Piłka wodna i wiele innych. Dziś natomiast trochę o Tenisie.

Dyscyplina ta jest nierozdzielnie związana z pieniędzmi. To, co zachęca najznamienitszych do gry, to olbrzymie nagrody. Tajemnicą jest konto Beckera, wiadomo już jednak, że na tenisie zarobił miliony dolarów. I nie tylko on jeden.

Serve & Volley jest oczywista ponad tym wszystkim. Należy skoncentrować się na grze i na niczym innym. Od razu też ostrzegam — trzeba mieć dużo czasu i umiejętności, by stać się naprawdę dobrym tenisistą — zarówno w rzeczywistości, jak i w Serve & Volley.

Na samym początku mamy możliwość skonfigurowania gry. Liczba setów i szybkość „biegania” piłki — to wbrew

pozorom dość ważne parametry. Dla dwóch straceńców jest gra podwójna, dla pojedynczych — singlowa lub turniejowa. Nie zapomniano też o praktyce (widać autorzy mieli odpowiednią praktykę).

Kort jak zwykle widziany jest z boku — ni to ułatwia, ni to utrudnia. Sędzia jest, ale tylko siedzi. Trybuny są, ale tylko krzyczą. Widać też przeciwnika, który czasem gra tak, jakby chciał ci pomóc, a czasem szaleje niczym Zawodowiec. Taki już jest ten Wimbledoński folklor.

Serwis i odbicie piłki, czyli mówiąc ogólniej gra, zostały bardzo poważnie urozmaicone

W momencie przyciśnięcia FIRE na joysticku w rogu ponad graczem ukazuje się okno serwisowe. W nim jest ukazana siła uderzenia (zależna od zmęczenia zawodnika) oraz rodzaj możliwego serwisu — lekki, mocny i as. Wybór jednego z nich następuje poprzez ponowne naciśnięcie FIRE, gdy czerwona kreska, przesuwająca się do góry znajdzie się na odpowiedniej literce umieszczonej w oknie.

Zaraz potem ta sama kreska zaczyna z powrotem zjeżdżać w dół. Gdy tylko zetknie się z poziomą linią w samym środku skali, wciśnij momentalnie FIRE. Manewr ten służy do poprawnego uderzenia piłki — jeżeli powyższych czynności nie wykonasz dokładnie, serwis będzie niecelny.

Odbiór piłki jest podobny. W chwili gdy komputer włączy okno odbioru, identyczna jak przy serwisie kreska przesuwa się ku górze ekranu. Możesz wtedy zdecydować się na normalne odbicie, volley, smash lub lob. Każde z nich wymaga oczywiście zupełnie innej precyzji, gdy kreska zacznie opadać w dół. W przypadku smasha moment zetknięcia kreski z poziomą linią musi być wychwycony idealnie, a dla loba błąd przy wciśnięciu FIRE może być nawet bardzo duży.

Pamiętać też trzeba, że masz możliwość precyzyjnego kierowania piłki w różne części kortu. Zawsze gdy włączone jest okno, widać w nim, oprócz skali z poruszającą się kreską, część kortu. Niebieski kwadracik na nim wskazuje miejsce lotu piłki. Kierujesz nim za pomocą joysticka.

To już chyba wszystko, co można powiedzieć o Serve & Volley. Nie warto nawet wspominać o grafice czy muzyce, bo ich po prostu prawie nie ma. Duża natomiast jest satysfakcja z samej gry. Po raz pierwszy chyba gracz kieruje swoimi uderzeniami w sposób w pełni kontrolowany, a nie na zasadzie Toto-Łotka. Czynnikiem ten przyciągnie zapewne wielu, którym życzę polamania rakiet!

Lavender Blue

Firma: Accolade
Komputer: Spectrum, Commodore, Atari



P-47

Gdy byłem mały, chciałem zostać prawdziwym pilotem (a kto nie chciał?). Potem grałem na kilku symulatorach lotu i zniechęciłem się nieco. Niedawno zagrałem w "P-47" i zupełnie mi przeszło — nie jestem samobójcą.

P-47 to nazwa amerykańskiego myśliwca z II wojny światowej, powszechnie używanego do tępienia podwładnych Adolfa H. W tej grze zadanie jest dokładnie takie samo: zabić i rozwalić wszystko co się rusza i to, co się nie

rusza, strzelać seriami do wszystkiego w polu widzenia. Po twoim przelocie mogą zostać tylko dymiące wraki i dziury w ziemi. Do ataku!

Lecisz sobie spokojnie, a tu nagle wypada eskadra Messerschmittów. Parę uników, kilka serii i szkopy odbywają swoją ostatnią drogę, malowniczo płonąc. Jakież działo? *No problem.* Ty też chcesz sobie postrzelać! Leci helikopter (skąd helikopter w tych czasach?) — nie żałuj amunicji! Bombowce i czołgi to nieco większy problem, ale nie dla

ciebie! Nawet eskadry rakiet V-2 są tylko drobną przeszkodą w wykonaniu zadania. Nie warto nawet wspominać o burzy śnieżnej; to nie robi na tobie, krwiożerczym mordercy niewinnych, żadnego wrażenia.

Trafiony helikopter zamienia się w „coś takiego”, co należy zabrać. Dostaniesz wtedy dodatkowe uzbrojenie. Jest tego spory wybór: niekierowane rakiet, bomby „Tallboy” (przebijają 10 m żelbetu), sterowane działko „Oerlikon” (sprężone dwulufowe, kal. 20 mm), eksplodujące pociski, przyspieszacz czy dodatkowy samolot. Przyjemne jest to, że kiedy wrednym szwabom uda się podziurawić cię jak sito, nie tracisz dodatkowej broni wracając do gry, staje się ona tylko trochę słabsza (mniej rakiet w jednej salwie itp.).

Pierwszy etap to zwykły lot nad zupełnie nieciekawym terenem, kończący się spotkaniem z pociągiem pancernym. Nad jednym z wagonów jest bezpieczne miejsce, załóż tam i zrzuć bomby. Gdy pociąg zamieni się w kupę płonącego i zupełnie bezużytecznego złomu, przechodzisz do drugiego etapu. Tym razem jest to lot nad chmurami, z widokiem na wschodzące słońce (w kolorze wygląda wspaniale), zakończony spotkaniem z superbombowcem. Nie jest to B-17 i posiada imponujące „martwe pole” — ustaw się pod nim, poślij go w dół w płomieniach. Na dalszych poziomach spotkasz przerośnięte „Tygrysy”, następne bombowce-giganty (pomylili skalę) oraz dwa krążowniki, czekające na kogoś, kto je w końcu zatopi.

„P-47 Thunderbolt” (ciekawe, w gazetach nazywa się on „P-47 Thunderbolt”, a na dysku „P-47 Freedom Fighter”) nie jest może rewelacją, ale nie można mu odmówić dobrej jakości. Jest to porządna strzelanina, w której można wyładować mordercze instynkty bez popadania w konflikt z kodeksem karnym. W zalewie nędznych, ledwo-ledwo napisanych gier, „P-47” jest przykładem solidności — dopracowany w szczegółach, szybki i kolorowy. To jest to, co tygrysy lubią najbardziej. Zagraj, a polubisz „P-47”.

Glaurung

Komputer: Spectrum, Commodore, Amstrad



MONITOR BIAZET MMK 127

Dzięki uprzejmości zakładów UNITRA-BIAZET redakcja otrzymała do testowania monitor MMK 127 i telewizor-monitor TMP 205. W tym numerze przedstawiamy redakcyjny test monitora. W jednym z następnych ukaże się test telewizora-monitora.

Monitor został umieszczony w estetycznej obudowie z białego tworzywa sztucznego o wymiarach 33x31x30 cm. W tylnej części obudowy znajdują się otwory wentylacyjne, których wielkość zapewnia dobre chłodzenie układom elektronicznym i na pewno ma wpływ na jakość ich pracy.

Kształt przedniej ścianki pozwala na łatwą instalację filtra ochronnego za pomocą kupowanych z nim przylepców i rzepów. Filtr ochronny wyraźnie poprawia komfort pracy monitorem, likwidując uciążliwe refleksy świetlne od ekranu kineskopu.

U dołu przedniej ścianki umieszczono wszystkie podstawowe pokrętki regulacyjne, tzn.: jasności, kontrastu, głośności, a także wyłącznik sieciowy i diodę świecącą sygnalizującą włączenie zasilania. Wszystkie te elementy są wyraźnie i czytelnie opisane, zaś ich rozmieszczenie zapewnia wygodną pracę. Monitor został wyposażony w bezimpedycyjny kineskop o zielonej barwie świecenia i przekątnej ekranu 12". Zielona barwa nie męczy wzroku nawet podczas długotrwałej pracy z mikrokomputerem.

Na tylnej ścianie znajduje się gniazdo zasilania i gniazdo do podłączenia mikrokomputera. Zastosowano typowe gniazdo DIN, podobnie jak w monitorach Neptun 156 czy też VELA. Układy elektryczne wymagają podania na to gniazdo całkowitego sygnału wizyjnego i sygnału fonii. Ponieważ większość popularnych mikrokomputerów nie jest

wyposażona w gniazdo DIN, konieczne będzie przygotowanie przewodu połączeniowego.

Rozmieszczenie sygnałów w gnieździe jest opisane w instrukcji obsługi monitora, jednak powinny również znaleźć się tam dokładne opisy wykonania przewodu połączeniowego do popularnych mikrokomputerów. Nie każdy użytkownik komputera będzie w stanie samodzielnie wykonać przewód, tym bardziej że typowym złączem dla sygnału wizji w komputerze jest gniazdo „cinch”.

Po włączeniu monitora starałem się uzyskać optymalny obraz manipulując pokrętkami jasności i kontrastu. Zakres ich działania jest szeroki i z łatwością można taki obraz uzyskać. Jeśli jednak regulacji tych będziemy dokonywać przy ciemnym tle i jasnych literach tekstu, to po zmianie obrazu na negatywowo (o jasnym tle i ciemnych literach) będziemy zmuszeni ponownie ustawiać jasność i kontrast. Układ elektryczny nie zawiera bowiem układu odtwarzania składowej stałej sygnału wizyjnego. Jego brak powoduje, że średnia jasność obrazu pozostaje stała. Przy częstych zmianach jasności dużych części ekranu jest to niewygodne i zmusza do każdorazowego regulowania monitora.

Zniekształcenia geometryczne obrazu są niewielkie i praktycznie niezauważalne. Okrąg narysowany na ekranie miał wymiary w poziomie 115 mm, w pionie zaś — 113 mm. Tekst był wyraźny i kontrastowy. Ostrość, znakomita w środkowej części ekranu, pogarsza się w jego rogach. Tekst staje się nieco rozmyty, szczególnie przy dużej jasności świecenia ekranu. Zjawisko to jest typowe dla wszystkich monitorów przeznaczonych do popularnych mikrokomputerów i zbudowanych z podzespołów odbiornika TV.

Monitory profesjonalne o dużej rozdzielczości są obowiązkowo wyposażone w układ dynamicznego ogniskowania plamki, zapewniający ostry obraz na całym ekranie. Układ ten zmienia napięcie ogniskujące plamkę w zależności od położenia strumienia elektronów rysujących obraz.

Monitor MMK 127 bez kłopotu współpracował z komputerami ZX Spectrum i Atari 800. Dalszym krokiem było podłączenie go do komputera IBM PC z kartą CGA. Karta ta posiada wyjście całkowitego sygnału wizji (composite video) i bez problemu można ją podłączyć do monitora. Niestety, początkowo próba ta się nie powiodła. Obraz na ekranie był widoczny, lecz nie był zsynchronizowany w pionie. W testowanym monitorze nie wyprowadzono na zewnątrz potencjometrów synchronizacji, jak na przykład w VELL i monitorach profesjonalnych. Konieczne zatem stało się rozkręcenie obudowy. Elementy regulacyjne na płycie drukowanej są dokładnie opisane i z łatwością można je znaleźć. Po tej drobnej regulacji monitor bezbłędnie wyświetlał obraz z karty CGA, zarówno w trybie tekstowym, jak i graficznym.

Jakość obrazu można określić jako dobrą. Współpraca z kartą CGA przyda się z pewnością posiadaczom przenośnych komputerów osobistych z ciekłokrystalicznymi ekranami. W warunkach stacjonarnych praca z komputerem będzie znacznie wygodniejsza.

Jakość dźwięku reprodukowanego przez wewnętrzny głośnik była bardzo dobra. Monitor ma silny wzmacniacz sygnału fonii. Dla maksymalnej głośności dźwięk staje się nieznośny.

W redakcji monitor pracował przez dwa miesiące. W tym czasie nie wystąpiły praktycznie żadne usterki w jego pracy. Jedynym zauważonym mankamentem był pisk wydobywający się z układów odchylenia. Nie był on głośny, ale dokuczliwy. Pojawiał się sporadycznie i nieokresowo. Z przyjemnością zauważyłem, że aby się go pozbyć należy stuknąć pięścią w obudowę. Oczywiście za każdym razem skwapliwie to czyniłem. Na koniec należy w krótkim podsumowaniu wymienić podstawowe parametry techniczne oraz zauważone wady i zalety monitora.

PARAMETRY TECHNICZNE:

Rozdzielczość	640x200 punktów
Kineskop	12" zielony
Częstotliwość odchylenia poziomego	15625 Hz.
Częstotliwość odchylenia pionowego	50 Hz.
Maksymalna moc fonii	1 W
Moc pobierana z sieci	30 W
Masa	6,4 kg
Rozmiary	330x310x300 [mm]

ZALETY:

1. Estetyczna obudowa z dużą ilością otworów wentylacyjnych.
2. Łatwo dostępne i wyraźnie opisane elementy regulacyjne.
3. Prostota montażu filtra ochronnego.
4. Dobra jakość obrazu i dźwięku.

WADY:

1. Brak układu odtwarzania składowej stałej sygnału wizji.
2. Zła współpraca z kartą CGA.
3. Sporadyczny pisk z układów odchylenia.

**Producent: Białostockie Zakłady Podzespołów Telewizyjnych
UNITRA-BIAZET
Sz. Płn. Obwodowa 38
15-113 Białystok**

Kupno monitora MMK 127 można polecić wszystkim użytkownikom popularnych komputerów 8-bitowych i przenośnych komputerów osobistych. Zamiana telewizora na monitor nie tylko wyraźnie poprawi komfort pracy, ale i wyeliminuje konflikty rodzinne powstające przy wspólnym użytkowaniu telewizora. Monitor można kupić w sklepie fabrycznym zakładów BIAZET w Białymstoku za 700 tys. zł.

Robert Magdziak



dla każdego
cz. 1

Wszyscy użytkownicy Amstradów wyposażonych w stacje dysków znają komendy takie jak IDIR, IERA czy IREN. Użytkownicy CPC 464 znają inne podobne komendy, np. IBASIC, których cechą wspólną jest pionowa kreska (ang. bar) na początku. Są to właśnie RSX-y (Resident System eXtension — rezydentne rozszerzenie systemu, często nazywane bar-commands). Wiele programów instaluje RSX-y, rozszerzając tym sposobem wbudowany BASIC.

WSTĘP TEORETYCZNY

Czym jest RSX? Jest to procedura w kodzie maszynowym, identyfikowana przez NAZWĘ, a nie, jak zwykła procedura w kodzie, przez adres. Komu to potrzebne? Jest to pytanie sensowne, jeśli mamy kilka procedur, jeśli zaś mamy program rozszerzający BASIC, zawierający np. 70 procedur (są i większe), wtedy zapamiętanie tylu adresów jest (łagodnie mówiąc) kłopotliwe. Użycie nazwy "CIRCLE" zamiast adresu (np. &A472) jest po pierwsze: wygodniejsze, po drugie: łatwiejsze do zapamiętania (i zrozumienia).

SKĄD SIĘ BIORĄ RSX-Y?

Jak zainstalować RSX-a? Jak większość operacji na Amstradzie, jest to proste. Należy zacząć od stworzenia dwóch struktur. Pierwsza z nich — to tablica nazw komend. Składa się ona z nazw komend, pisanych dużymi literami. Ostatni bajt (litera) każdej nazwy musi mieć ustawiony 7. bit (przez wykonanie operacji OR 128 lub dodanie 128). Nazwy mogą zawierać litery A-Z (tylko łacińskie), kropkę i cyfry, jednak żadna nazwa nie może zaczynać się od cyfry. Tablicę kończy bajt o wartości zero. Druga struktura — to tablica skoków. Składa się ona ze słowa (dwa bajty) zawierającego adres tablicy nazw oraz skoków do procedur (kodu) RSX-a (tzw. długich, mnemonik "JP adr"). Każdej nazwie odpowiada jeden skok do procedury; skoki muszą być umieszczone w tablicy w takiej samej kolejności jak nazwy. Trzeba jeszcze tylko zarezerwować cztery bajty na dane o przyłączeniu do kolejki. Teraz użyjemy procedury KL LOG EXT o adresie #BCD1. W parze rejestrów BC umieszczamy adres tablicy skoków, w parze HL — adres tych właśnie czterech bajtów i wykonujemy CALL #BCD1. Operacja ta nie sygnalizuje błędu, nawet jeśli dane są błędne. Są tylko dwa zastrzeżenia. Pierwsze: wszystkie trzy struktury RSX-a (tablica nazw, tablica skoków, bajty na podłączenie do kolejki) muszą się znajdować w obszarze od &4000 do &C000. Drugie: długość nazwy nie powinna przekraczać szesnastu znaków.

GDZIE SĄ MOJE PARAMETRY!?

Przy wywołaniu RSX-a (i tak samo przy rozkazie "CALL" w BASIC-u) możemy przekazać procedurze parametry (piszemy: "IRSX,parametr,parametr..."). Na wejściu do procedury akumulator zawiera liczbę parametrów (od 0 do 31). Można więc łatwo stwierdzić, ile parametrów przekazano i czy jest to właściwa ilość. Same parametry wskazywane są przez rejestr IX. Rejestr ten wskazuje na ostatni parametr, poprzedni znajduje się pod adresem IX+2, trzeci od końca pod adresem IX+4 i tak dalej, n-ty od końca parametr znajduje się pod adresem IX+(n-1)*2.¹⁾ Jeżeli potrzebujemy tylko jednego parametru, możemy sobie ułatwić życie: para DE zawiera wartość ostatniego parametru. Każdy parametr ma postać 16-bitowego słowa (czyli są to dwa bajty). W przypadku liczb całkowitych jest to po prostu liczba (dopuszczalny jest zakres -32768 do 32767 lub 0 do 65535). Liczby rzeczywiste przekształcane są na całkowite (funkcją INT); jeżeli przekraczają dopuszczalny zakres, to wystąpi błąd "Overflow". Z łańcuchami jest trochę kłopotu. Do procedury przekazywany jest adres tzw. deskryptora. Deskryptor składa się z trzech bajtów: bajtu długości (0 do 255) oraz słowa zawierającego adres łańcucha. W przypadku CPC 464 kłopot jest podwójny: nie można łańcucha przekazać bezpośrednio (np. IDIR, "PROG*.*"), trzeba go przekazać jako zmienną (a\$="PROG*.*"IDIR,@a\$).

MAŁY PRZYKŁADZIK

Listing 1 przedstawia przykładowy RSX w asemblerze (Laser Genius), listing 2 to RSX po asemblacji i dodaniu loadera w BASIC-u. Ten przykład demonstruje definiowanie RSX-a oraz przekazywanie parametrów do RSX-a.

Definiowane są dwie komendy RSX: IHIGH.PRINT,x,y,@text\$ IHIGH.SET,x
Pierwsza (HIGH.PRINT) drukuje łańcuch literarii o zmienionej wysokości, zaczynając od współrzędnych (x,y). Druga komenda (HIGH.SET) ustala wysokość liter — od 1 do 24.

IDZIEMY Z POSTĘPEM!

Listing 3 to program definiujący komendy RSX (tworzący procedurę instalacyjną). Jeśli mamy kilka procedur w kodzie maszynowym, to czemu nie zrobić z nich RSX-ów?

Swoje procedury należy załadować do pamięci PRZED uruchomieniem programu "Make RSX", a potem wystarczy tylko odpowiadać na pytania i zapisać nowe parametry...

OSTRZEŻENIE: Jeśli program używa więcej niż dwóch procedur w kodzie maszynowym, muszą być one zdefiniowane jako RSX-y. Kto się nie podporządkuje, naraża się na prześladowania. (Nie będzie litości!).

C.D.N.

Ciąg dalszy nastąpi, a w nim: jak przekazać wyniki pracy RSX-a do BASIC-a, jak wywołać RSX-a z poziomu kodu maszynowego oraz dalsze przykłady.

Michał Szokoło

¹⁾ Stąd można wyliczyć, że jeśli podajemy x parametrów, to parametr nr y (zał. y<=x) jest pod adresem IX+(x-2-y)*2. Wzór jest prawdziwy dla parametrów numerowanych od 1, gdy numerujemy parametry od zera, wzór zmienia się na IX+(x-y)*2.

Bibliografia:
1. Brückmann, Englisch, Gerits i Steigers "CPC Intern"
2. D. Thomasson "The Ins & Outs of the Amstrad"

```
10 #LIST OFF
#COUNT ON
; RSX dla kazdego - Czesć I
; Copyright 1990 by SEM 03375991 B
;
; ORG #A600
20 instal:
LD BC,tablica
LD HL,kolejka
JP KL.LOG.EXT
kolejka:
DS 4
tablica:
DW nazwy
JP hiprint
JP hi.set
nazwy: DB "HIGH.PRIN","I"+128
DB "HIGH.SE","I"+128
DB 0
30 ; zmienne
cy : DB 0 ; wspolrzedna y
cx : DB 0 ; wspolrzedna x
dlug : DB 0 ; dlugosc lancucha
tekst: DW 0 ; adres lancucha
adres: DW 0 ; adres na ekranie
adresl:
DW 0 ; kopia zmiennej "adres"
znak : DB 0 ; kod znaku do druku
wys : DB 2 ; wysokosc znaku
40 hiprint:
; poczatek
CP 3
RET NZ ; czy trzy parametry?
LD A,(IX+2) ; wez drugi parametr
LD (cy),A
LD A,(IX+4) ; wez pierwszy parametr
LD (cx),A
EX DE,HL ; wez ostatni parametr
LD A,(HL) ; odczytaj deskryptor
INC HL
LD E,(HL)
INC HL
LD D,(HL)
LD (tekst),DE ; adres tekstu
DEC A
RET C ; czy pusty?
ADD A,2
LD (dlug),A ; zapamietaj dlugosc
50 ; ustalamy parametry systemowe
LD HL,(cy) ; wspolrzedne std.
DEC H ; na wewnetrzne
LD L,(cy),HL
60 kolejny:
LD A,(dlug)
DEC A
RET Z ; lancuch pusty?
LD (dlug),A
LD HL,(tekst) ; wez znak
LD A,(HL)
LD (znak),A ; zapamietaj znak
INC HL
LD (tekst),HL ; przesun wskaznik
CALL HIKL.LOW.ON ; wlacz dolny ROM
LD HL,(cy) ; wez pozycje znaku
PUSH HL
INC H ; x:=x+1
LD (cy),HL ; zapamietaj
POP HL
PUSH HL
LD A,(znak)
CALL DISPLAY ; wydrukuj znak
CALL HIKL.LOW.OFF ; wylacz dolny ROM
POP HL
CALL SCR.CHAR.POS ; wez adres pola znaku
LD (adres),HL ; zapamietaj
LD C,B ; szerokosc znaku do C
70 ; petle powiekszania znaku
pozio:
LD HL,(adres) ; wez adres na ekranie
LD (adresl),HL ; zrob kopie
PUSH HL
CALL w.prawo ; przesun o bajt w prawo
LD (adres),HL ; zapamietaj
POP HL
LD DE,bufor
LD B,B
zapam: LD A,(HL) ; zapamietaj kolumne
LD (DE),A
CALL w.dol
INC DE
DJNZ zapam
LD HL,(adresl) ; wez adres pola
LD DE,bufor ; znaku i bufora
LD B,B ; osiem linii
duzy : PUSH BC
LD A,(wys) ; kazda linia
LD B,A ; "wys" razy
duzy1: LD A,(DE)
LD (HL),A
CALL w.dol
DJNZ duzy1
INC DE
POP BC
DJNZ duzy
DEC C ; nastepna kolumna?
JR NZ,pozio
JP kolejny ; kolejny znak
80 ; przesuniecie wskaznika na ekranie
w.dol: PUSH DE
PUSH BC
CALL SCR.NEXT.LINE
POP BC
POP DE
RET
w.prawo:
PUSH DE
PUSH BC
CALL SCR.NEXT.BYTE
POP BC
POP DE
RET
90 ; ustawianie wysokosci znaku
hi.set:
CP 1 ; czy 1 parametr?
RET NZ
LD A,E ; wez młodszy bajt
OR A
RET Z ; koniec, jesli zero
CP 25
RET NC ; koniec jesli za duzy
LD (wys),A ; ustaw wysokosc
RET
100 ; adresy procedur systemowych
KL.LOG.EXT:
EQU #BCD1
SCR.NEXT.LINE:
EQU #BC26
SCR.NEXT.BYTE:
EQU #BC20
SCR.CHAR.POS:
EQU #BC1A
HIKL.LOW.ON:
EQU #B906
HIKL.LOW.OFF:
EQU #B909
DISPLAY:
EQU #BDD3
bufor: DS 8
```

LISTING 2

```
1000 '
1010 ' RSX dla kazdego - Czesć I
1020 ' (c)1990 by SEM 03375991 B
1030 '
1040 GOTO 1210
1050 DATA 01,0d,a6,21,09,a6,c3,d1,bc,00,00,00,00,15,a6,c3
1060 DATA 33,a6,c3,d5,a6,48,49,47,48,2e,50,52,49,4e,d4,48
1070 DATA 49,47,48,2e,53,45,d4,00,00,00,00,00,00,00,00
1080 DATA 00,00,02,fe,03,c0,dd,7e,02,32,28,a6,dd,7e,04,32
1090 DATA 29,a6,eb,7e,23,5e,23,56,ed,53,2b,a6,3d,d8,c6,02
1100 DATA 32,2a,a6,2a,28,a6,25,2d,22,28,a6,3a,2a,a6,3d,c8
1110 DATA 32,2a,a6,2a,2b,a6,7e,32,31,a6,23,22,2b,a6,cd,06
1120 DATA b9,2a,28,a6,e5,24,22,28,a6,e1,e5,3a,31,a6,cd,d3
1130 DATA b0,cd,09,b9,e1,cd,1a,bc,22,2d,a6,48,2a,2d,a6,22
1140 DATA 2f,a6,e5,cd,cd,a6,22,2d,a6,e1,11,e2,a6,06,08,7e
1150 DATA 12,cd,c5,a6,13,10,f6,2a,2f,a6,11,e2,a6,06,08,c5
1160 DATA 3a,32,a6,47,1a,77,cd,c5,a6,10,f9,13,c1,10,f0,0d
1170 DATA 20,ca,c3,5b,a6,d5,c5,cd,26,bc,c1,d1,c9,d5,c5,cd
1180 DATA 20,bc,c1,d1,c9,fe,01,c0,7b,b7,c8,fe,19,d0,32,32
1190 DATA a6,c9,00,00,00,00,00,00,00,00,00,00,00,00
1200 DATA 8328
1210 SYMBOL AFTER 256 : MEMORY &A5FF
1220 RESTORE 1050 : sum=0
1230 FOR ADR=&A600 TO &A6E9
1240 READ a$
1250 b=VAL("&"a$)
1260 sum=(sum+b)MOD &4000
1270 POKE adr,b
1280 NEXT adr
1290 READ b
1300 IF b<>sum THEN PRINT"DATA ERROR!" : STOP
1310 CALL &A600 : POKE &A600,&c9
1320 MODE 1 : CALL &BC02 : PEN 1 : PAPER 0
1330 !HIGH.SET,2
1340 CLS : LOCATE 1,3
1350 !HIGH.PRINT,1,1," !HIGH.PRINT RSX (c)1990 SEM 03375991 B"
1360 NEW
```


CP/M Plus

procedury systemowe w Turbo Pascalu (cz. 1)

W przypadku praktycznie każdego komputera dostęp do niektórych zasobów sprzętowych lub informacje o nich nie są osiągalne w prosty sposób z poziomu języka programowania.

Dotyczy to na przykład zegara systemowego lub danych o rozmiarze pamięci operacyjnej liczbie napędów operacyjnego systemu operacyjnego.

Dostęp do tych informacji wymaga dobrej znajomości dokumentacji sprzętu i jest realizowany w różny sposób na różnych komputerach. Prezentowany cykl procedur systemowych w języku Turbo Pascal ma ułatwić to zadanie dla 8-bitowych komputerów Amstrad.

W części pierwszej przedstawiamy 10 użytecznych procedur (LISTING 1. Plik

SYS1.SYS). Większość informacji pozwalających na ich napisanie była wcześniej publikowana na łamach Bajtka. Obecnie zebrano je w jednym miejscu w zwięzłej i przejrzystej formie, umożliwiającej łatwe wykorzystanie we własnych programach.

Pierwsze trzy procedury i funkcje (UnTOD, Time i Date) pozwalają na dostęp do zegara systemowego. Funkcje FDD, RAM, SIO i PCW określają zasoby sprzętowe używanego komputera. Pozostałe trzy podają wersję systemu operacyjnego, numer seryjny systemu i numer wersji BIOS-u.

Demonstrację działania zestawu procedur jest program SystemDemo1, prezentowany na listingu 2. Efekt działania programu przedstawia rys. 1.

Jonasz Mayer

LISTING 1 Plik SYS1.SYS

```
(*****)
(*      Plik SYS1.SYS. (C) JM 1989      *)
(*      *)
(*      Biblioteka procedur systemowych w języku Turbo Pascal *)
(*      przeznaczonych na komputery Amstrad PCW 8256/8512 i *)
(*      CPC 6128. Cześć pierwsza,      *)
(*      *)
(*      Lista procedur i funkcji:      *)
(*      *)
(*      1. Procedure UnTOD (Var time,date : DTstr; TOD : TODrec); *)
(*      2. Function Time : DTstr;      *)
(*      3. Function Date : DTstr;      *)
(*      4. Function GetVersionNo : DTstr; *)
(*      5. Function GetSerialNo : DTstr; *)
(*      6. Function FDD : byte;      *)
(*      7. Function RAM : integer;      *)
(*      8. Function SIO : boolean;      *)
(*      9. Function PCW : boolean;      *)
(*      10. Function BIOSVersion : DTstr; *)
(*      *)
(*****)
const
  getTOD = $69; { Numer funkcji BDOS }
                { pobierające czas i date }

type
  TODrec = record
    n_day : integer; { dni od 1 stycznia 1978 }
    hour : byte;     { godzina w formacie BCD }
    min : byte;      { minuty w formacie BCD }
    sec : byte;      { sekundy w formacie BCD }
  end;
  DTstr = string(8);

Procedure UnTOD (Var time,date : DTstr; TOD : TODrec);
(*****)
(* Procedura dekoduje czas i date z 4-bajtowej struktury TOD, *)
```

```
(* Stosowana przy funkcjach Time i Date, *)
(*****)
const
  months : array(1..12) of byte = (31,28,31,30,31,30,
                                    31,31,30,31,30,31);

var
  days : integer;
  year,month,day,flag : byte;
  d1,d2,m1,m2,y1,y2 : byte;
  s1,s2,h1,h2 : byte;

begin
  with TOD
  do begin
    (* wyznaczenie godzin, minut i sekund *)
    h1 := hour shr 4; h2 := hour and $0F;
    m1 := min shr 4; m2 := min and $0F;
    s1 := sec shr 4; s2 := sec and $0F;
    time := chr(48+h1)+chr(48+h2)+':'
            +chr(48+m1)+chr(48+m2)+':'
            +chr(48+s1)+chr(48+s2);
    end;
  days := TOD.n_day;

  year := 78; { 1978 1 Styczen }
  repeat
    if year mod 4 = 0 then flag := 1 else flag := 0;
    days := days - 365 - flag;
    year := year + 1;
  until days < 0;
  year := year - 1;

  (* wyznaczn miesiąc *)
  month := 12;
  repeat
    days := days + months(month);
    if month=2 then days := days + flag;
    month := month - 1;
  until days > 0;
  month := month + 1;

  (* wyznaczn dzień *)
  day := days;

  d1 := day div 10; d2 := day mod 10;
  m1 := month div 10; m2 := month mod 10;
  y1 := year div 10; y2 := year mod 10;
  date := chr(48+d1)+chr(48+d2) +
           ':' + chr(48+m1)+chr(48+m2) +
           ':' + chr(48+y1)+chr(48+y2);
end; (* of UnTOD *)

function time : DTstr;
(*****)
(* Zwraca czas zegara systemowego, *)
(*****)
var t,d : DTstr; a : byte;
    TOD : TODrec;
begin
  A := BDOS (GetTOD,addr(TOD));
  TOD.sec := a;
  UnTOD (t,d,TOD);
  time := t;
end; (* of TIME *)

Function Date : DTstr;
(*****)
(* Zwraca date zegara systemowego, *)
(*****)
var t,d : DTstr;
    TOD : TODrec;
begin
  BDOS (GetTOD,addr(TOD));
  UnTOD (t,d,TOD);
  Date := d;
end; (* of Date *)

function FDD : byte;
(*****)
(* Zwraca liczbę napędów dysków elastycznych, *)
(*****)
var a : byte;
begin
  InLine ($CD/$5A/$FC/$E6/$00/$32/a);
  if a=0 then FDD := 1 else FDD := 2;
end; (* of FDD *)

Function RAM : integer;
(*****)
(* Zwraca liczbę kilobajtów pamięci RAM, *)
(*****)
var bc : integer;
begin
  InLine ($CD/$5A/$FC/$E6/$00/$ED/$43/bc);
  RAM := 16 * Hi(bc);
end; (* of RAM *)

Function SIO : boolean;
(*****)
(* Ustala czy w systemie zainstalowano złącze szeregowe RS232, *)
(*****)
var bc : integer;
begin
  InLine ($CD/$5A/$FC/$E6/$00/$ED/$43/bc);
  SIO := (Lo(bc)=$FF);
end; (* of SIO *)

Function PCW : boolean;
(*****)
(* Ustala typ komputera (PCW lub CPC), *)
(*****)
```


PROCEDURY SYSTEMOWE
AMSTRADA (cz. III)

PAKIET SYSTEMU OPERACYJNEGO
(KERNAL) — dokończenie

```
(*****)
var a : byte;
begin
  inline ($CD/$5A/$FC/$E3/$00/$32/a);
  PCW := (a=1);
end; (* Version *)

Function GetVersionNo : DTstr;
(*****)
(* Zwraca wersje systemu operacyjnego. *)
(*****)
var HL : integer;
begin
  HL := BDOSHL ($0C);
  case HL
  of $0031 : GetVersionNo := 'CP/M 3.0';
    $0022 : GetVersionNo := 'CP/M 2.2';
    $0000 : GetVersionNo := 'CP/M 1.4';
    $0130 : GetVersionNo := 'MP/M II';
    $0222 : GetVersionNo := 'CP/NET';
  end;
end; (* of Get Version Number *)

Function GetSerialNo : DTstr;
(*****)
(* Zwraca numer seryjny systemu (zwykle 654321). *)
(*****)
var d : DTstr;
begin
  BDOS ($68,addr(d)+1);
  d[0] := chr(6);
  GetSerialNo := d;
end; (* of Get Serial No *)

Function BIOSVersion : DTstr;
(*****)
(* Zwraca numer wersji modulu BIOS systemu operacyjnego. *)
(*****)
var bc : integer;
s,t : string[5];
begin
  inline ($CD/$5A/$FC/$E3/$00/$ED/$43/bc);
  str (Hi(bc);1,s); str (Lo(bc);1,t);
  BIOSVersion := s + ',' + t;
end; (* of BIOS Version *)

(*****)
```

LISTING 2 Plik DEMOSYS1. PAS

```
Program SystemDemo1;
(*****)
(* Plik DEMOSYS1.PAS (C) JM 1988 *)
(* Demonstruje dzialanie biblioteki systemowej SYS1.SYS. *)
(* *)
(*****)

($I SYS1,SYS )

begin
  ClrScr;
  Write ('System operacyjny: ',GetVersionNo;8);
  if GetVersionNo<'CP/M 3.0'
  then halt;
  writeln (Date;20,Time;10);
  writeln ('Numer seryjny: ',GetSerialNo;8);
  writeln ('Wersja modulu BIOS:',BIOSVersion;8); writeln
  if PCW
  then writeln ('Amstrad PCW 8256/8512')
  else writeln ('Amstrad CPC 6128'); writeln;
  writeln ('Pamiec RAM [KB]: ',RAM;4);
  writeln ('Liczba napadow: ',FDD;4);
  if SIO
  then writeln ('RS 232C: ',1;4)
  else writeln ('Brak RS 232C');
end;
(*****)
```

```
System operacyjny: CP/M 3.0      03.10.89  05:59:25
Numer seryjny:      654321
Wersja modulu BIOS:      1.4

Amstrad PCW 8256/8512

Pamiec RAM [KB]:  512
Liczba napadow:   2
RS 232C:          1
```

Rys. 1 Efekt wykonania programu DEMOSYS1. PAS,

KL SYNC RESET — KASUJ ZDARZENIA SYNCHRONICZNE #BCF5

Kasuje kolejke zdarzen synchronicznych.
Wejscie: nie ma.
Wyjscie: AF i HL skasowane.

KL DEL SYNCHRONOUS — SKASUJ ZDARZENIE SYNCHRONICZNE #BCF8

Usuń zdarzenie synchroniczne z kolejki. Wejscie: HL zawiera adres bloku zdarzenia. Wyjscie: AF, BC, DE i HL skasowane.

KL NEXT SYNC — WEŻ NASTĘPNE ZDARZENIE SYNCHRONICZNE #BCFB

Pobranie następnego zdarzenia synchronicznego z kolejki. Wejscie: nie ma. Wyjscie: CY=0 jeśli nie ma zdarzenia do pobrania, wtedy A oraz HL skasowane. CY=1 jeśli jest zdarzenie do pobrania, wtedy HL zawiera adres bloku, A zawiera priorytet poprzedniego zdarzenia (jeśli było). DE zawsze skasowane.

KL DO SYNC — OBSŁUŻ ZDARZENIE SYNCHRONICZNE #BCFE

Wykonaj procedurę przypisaną do zdarzenia. Wejscie: HL zawiera adres bloku zdarzenia. Wyjscie: AF, BC, DE i HL skasowane.

KL DONE SYNC — ZAKOŃCZ OBSŁUGĘ ZDARZENIA SYNCHRONICZNEGO #BD01

Zakończenie obsługi zdarzenia synchronicznego. Wejscie: C zawiera priorytet poprzedniego zdarzenia, HL zawiera adres bloku zdarzenia. Wyjscie: AF, BC, DE i HL skasowane.

KL EVENT DISABLE — ZABLOKUJ ZDARZENIA SYNCHRONICZNE #BD04

Blokuje „zwykłe” (nie-ekspresowe) zdarzenia synchroniczne (czął kolejke). Wejscie: nie ma. Wyjscie: HL skasowane.

KL EVENT ENABLE — ODBLOKUJ ZDARZENIA SYNCHRONICZNE #BD07

Działanie odwrotne do KL EVENT DISABLE. Parametry jak KL EVENT DISABLE.

KL DISARM EVENT — ZABLOKUJ ZDARZENIE #BD0A

Zablokuj zdarzenie synchroniczne (nie zezwalaj na jego wystąpienie). Wejscie: HL zawiera adres bloku zdarzenia. Wyjscie: AF skasowane.

KL TIME PLEASE — ODCZYT CZASU #BD0D

Odczytuje czas zegara systemowego. Wejscie: nie ma. Wyjscie: DE/HL zawiera czas w 1/300 sek. W DE bardziej znaczące słowo.

KL TIME SET — USTAWIANIE CZASU #BD10

Ustawienie zegara systemowego. Wejscie: DE/HL zawiera nowy czas (w DE bardziej znaczące słowo). Wyjscie: AF skasowane. (MACHINE CODE) PAKIET KODU MASZYNOWEGO Pakiet kodu maszynowego (MC) składa się z dwóch procedur uruchomieniowych, procedur obsługi drukarki, CRTC (sterownika wizji) oraz PSG (generatora dźwięku).

MC BOOT PROGRAM — WCZYTAJ I WYWOŁAJ PROGRAM #BD13

Ładuj i wykonaj program. Wejscie: HL zawiera adres procedury ładowania i uruchomienia programu. Wyjscie: w zasadzie nie ma. Jeśli wystąpił błąd ładowania, to procedura ładująca powinna skasować CARRY. Jeśli program został załadowany poprawnie, to albo powinien zostać wykonany, albo procedura ładująca powinna wrócić z CY=1 i HL zawierającym adres startu.

MC START PROGRAM — WYWOŁAJ PROGRAM Z RAM LUB ROM #BD16

Wywołaj program znajdujący się w pamięci RAM lub ROM. Wejscie: nie ma (wykonanie programu). Uwaga: stos jest kasowany. Wejscie: HL zawiera adres startu, C zawiera kod ROM.

MC WAIT FLYBACK — CZEKAJ NA POWRÓT PROMIENIA #BD19

Czeka na powrót promienia elektronów (zakończenie tworzenia obrazu). Wejscie: nie ma. Wyjscie: wszystkie rejestry zachowane.

MC SET MODE — USTAW TRYB WYSWIETLANIA #BD1C

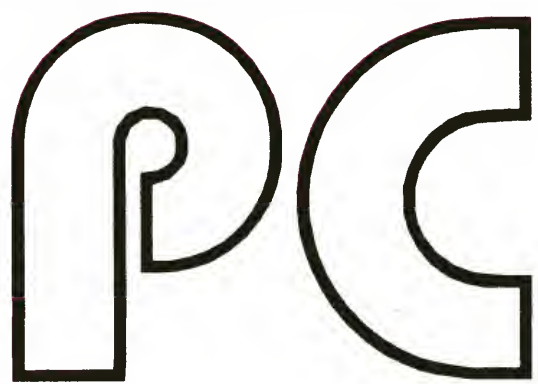
Ustawia tryb wyświetlania (jak SCR SET MODE #BC0E) ale bez czyszczenia ekranu. Wejscie: A zawiera nowy tryb (0—2). Wyjscie: AF skasowane.

MC SCREEN OFFSET — USTAW LOKALIZACJĘ EKRANU #BD1F

Ustawia lokalizację pamięci ekranu (baza i przesunięcie). Wejscie: A zawiera starszy bajt adresu pamięci ekranu, HL zawiera przesunięcie (0-16383). Wyjscie: AF skasowane. W następnym odcinku: dokończenie pakietu MC oraz procedury dodatkowe dla CPC 664 i CPC 6128.

Michał Szokoło

elementarz



Każdy komputer można rozebrać na części składowe. O ile jednak w przypadku Spectrum liczba elementów, które możemy po półgodzinnej pracy z lutownicą w prawej i śrubokrętem w lewej ręce ułożyć na stole, jest stała (nie licząc ułamanych nóżek itp.), o tyle w przypadku IBM PC części raz będzie więcej, raz mniej (nawet jeśli ograniczymy się do XT). Dlaczego tak się dzieje i co jest w środku — postaram się wyjaśnić w tym artykule.

Przy konstruowaniu PC przyjęto założenie, by umieścić możliwie największą ilość elektroniki potrzebnej do obsługi urządzeń zewnętrznych w sposób umożliwiający jej łatwą wymianę. Nie chodziło bynajmniej o ułatwienie napraw, ale o możliwość prostego dołączenia do komputera nowych urządzeń: stacji dysków, modemów, sieci i tak dalej. Zrealizowano to w następujący sposób — podstawową częścią komputera jest

płyta główna,

czyli duży obwód drukowany, na którym znajdują się: procesor, pamięć, kilka układów potrzebnych do wspomagania pracy procesora i kilka gniazd (tak zwanych slotów) na karty rozszerzające. Oprócz tego znajduje się na płycie głównej ROM-BIOS, czyli pamięć z zestawem podprogramów służących do wykonywania najbardziej podstawowych operacji na na razie jeszcze nieobecnych urządzeniach zewnętrznych. Jest to możliwe dzięki bardzo precyzyjnemu opisowi sposobów komunikacji z tymi urządzeniami. Do ich bezpośredniej obsługi, nie programowej, ale sprzętowej, służą

karty rozszerzające.

Te karty to nic innego jak odpowiednio przycięte obwody drukowane. Karty wkłada się w sloty złączem krawędziowym (analogicznym do tego wystającego z tyłu Spectrum). Dzięki odpowiedniemu umieszczeniu gniazd na płycie głównej, jedna ze stron kart zawsze jest dostępna na zewnątrz obudowy i można na niej umieścić gniazda pozwalające na podłączenie kabli do urządzeń zewnętrznych. Ze względu na różnice w budowie AT i XT, przed włożeniem karty do komputera trzeba sprawdzić, czy może ona w nim pracować. Niektóre karty mają przełącznik wybierający typ komputera, inne nie, co zwykle znaczy, że jest

im wszystko jedno, gdzie się znajdują.

Jakie jest niezbędne minimum kart rozszerzających? Na ogół potrzebne są co najmniej dwie — karta graficzna i kontroler stacji dysków elastycznych. Zaczniemy od tej pierwszej.

Karta graficzna

pozwala na przedstawienie wyników pracy komputera na ekranie monitora (a weźcie pod uwagę, że z punktu widzenia komputera nawet komunikat o niemożności uruchomienia systemu jest efektem jego pracy). Karty graficzne z jednej strony otrzymują od komputera informacje o tym, co ma się znaleźć na ekranie (informacje te są zapisywane w odpowiednim obszarze pamięci, znajdującym się na karcie), z drugiej strony generują na podstawie tych informacji obraz na monitorze. Istnieje kilka typów takich kart, różniących się między sobą rozdzielczością i ilością kolorów osiągalnych na ekranie, dlatego też porównywanie możliwości graficznych komputera PC z innymi komputerami jest trudne, gdyż nie wiadomo, co przyjąć za standard możliwości graficznych PC-ta.

W trybie tekstowym, w którym na ekranie mogą się pojawiać tylko litery i znaki semigraficzne (ramki, kilka liter greckich i symboli matematycznych), wszystkie karty proponują jako standard 25 wierszy po 80 znaków. Tryby graficzne to temat rzeka. Najtańsza i najbardziej popularna w Polsce karta monochromatyczna Hercules pozwala na pracę z rozdzielczością 720 na 348 punktów w dwóch kolorach i doskonale nadaje się do pracy z edytorami tekstu i grafiką nie wymagającą koloru. Druga popularna karta to CGA. Można z niej wycisnąć 4 kolory przy rozdzielczości 320 na 200 lub 2 kolory i 640 na 200 punktów. Nadaje się głównie do gier albo raczej nie nadaje się do niczego. Trzecia karta to EGA — można z niej wycisnąć 640 na 350 pun-

któw i 16 kolorów, czwarta — VGA — w zależności od producenta i wariantu daje od 640 na 480 punktów przy 16 kolorach, do 1024 na 768 punktów i 256 kolorów (z palety 262144 kolorów). VGA ma jednak sens dopiero przy korzystaniu z komputera AT lub szybszego, do XT nie warto kupować karty bardziej zaawansowanej niż EGA — ilość czasu potrzebnego do obsługi VGA może doprowadzić użytkownika do szału.

Do kontroli stacji dyskiek potrzebna jest następna karta. Jej możliwości są zwykle nieco obszerniejsze niż sama współpraca z napędami, dlatego też nazwą najpopularniejszego rozwiązania jest

Multi I/O

(czyt. multi aj-ou, bo to z angielskiego). *Multi* to wiele, litery I/O oznaczają Input/Output, czyli wejście — wyjście. I rzeczywiście, karta Multi I/O może służyć do obsługi (oprócz dwóch stacji dyskiek) jeszcze przynajmniej dwóch urządzeń zewnętrznych — jednego podłączonego przez łącze równoległe (tzw. Centronics, stanowiący również element karty graficznej Hercules), drugiego przez łącze szeregowe (RS-232). Z grubsza rzecz biorąc, pozwala to na podłączenie drukarki i myszy (lub drugiego komputera w celu ułatwienia przenoszenia danych między komputerami). Dodatkowo na kartach Multi I/O umieszczane są Game Port — czyli gniazdko do podłączenia joysticka i zegar baterijny.

Ponieważ klawiatura, jako jedyne urządzenie zewnętrzne, przyłączana jest bezpośrednio do płyty głównej, można by już właściwie zaczynać pracę, ale do pełnej wygody przydałby się jeszcze twardy dysk (przykręcony gdzieś wewnątrz obudowy) i

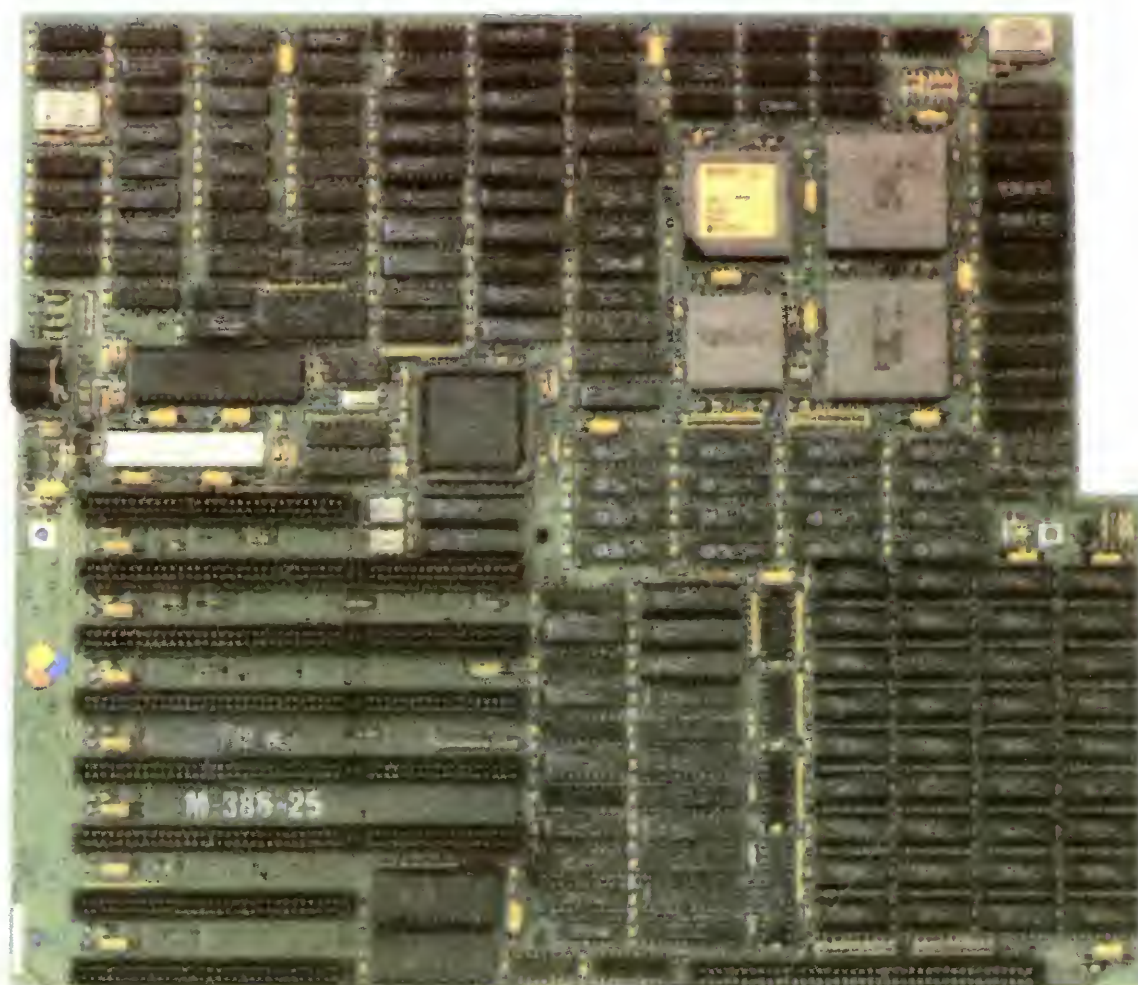
kontroler twardego dysku.

Ta karta pełni w stosunku do twardego dysku podobne funkcje jak karta Multi I/O przy dyskach elastycznych. Istnieje jednak między nimi pewna ciekawa różnica. Programy służące do obsługi dysków elastycznych znajdują się w ROM-ie na płycie głównej, natomiast te służące do obsługi dysku twardego są częścią kontrolera, dlatego też od niego w bardzo dużym stopniu zależy

szybkość działania twardego dysku. Własny ROM występuje dość często na kartach rozszerzających — dotyczy to przede wszystkim kontrolerów twardych dysków i kart EGA oraz VGA. Istnieją bardzo różne kontrolery, a dwa podstawowe parametry, umożliwiające ich rozróżnianie, to typ komputera — XT czy AT, i metoda zapisu informacji na dysku — MFM lub RLL (są i inne). Nie wnikając w szczegóły, ta druga pozwala na upakowanie na dysku o około 50% informacji więcej, ale nie każdy dysk pozwala na jej wykorzystanie.

W tym momencie skończyliśmy przegląd standardowego wyposażenia PC-eta. Przepraszam — został jeszcze koprocessor arytmetyczny. Ponieważ miesiąc temu zajmowaliśmy się koprocessorem dość dokładnie, tym razem tylko kilka zdań. Koprocessor to taki układ scalony, który wykonuje bardzo szybko różne operacje zmiennoprzecinkowe, czyli na liczbach rzeczywistych (procesor liczy tylko na liczbach całkowitych). Gniazdo pod koprocessor znajduje się na ogół na płycie głównej, tuż obok gniazda procesora. Ponieważ jednak istnieją koprocessory nie pasujące do tego gniazda (np. Weitek Abacus), stworzono specjalne karty, na których można te nietypowe koprocessory zainstalować.

Dla cierpiących na brak pamięci (w komputerze, nie w życiu prywatnym) opracowano karty umożliwiające zwiększenie pamięci operacyjnej powyżej 640 KB. Jest to tzw. standard EMS, pozwalający na dodatkowe zamontowanie do 8 MB pamięci, jednak nie wszystkie programy użytkowe potrafią z niej korzystać. Pamięć w PC-ta to zresztą osobny temat. Zastosowane rozwiązania ograniczają ilość dostępnej pamięci operacyjnej, z góry określając, pod jakimi adresami mają znajdować się pamięć ekranu, BIOS i ewentualny wbudowany interpreter BASIC-a (ten ostatni ze względu na zastrzeżenia patentowe występuje tylko w oryginalnych komputerach produkcji IBM). 640 KB zostawione użytkownikom wydawało się być w trakcie projektowania PC obszarem olbrzymim, praktycznie przez nich niewykorzystywalnym. Niestety, nie doceniono tempa rozwoju oprogramowania, które zajmowało coraz więcej pamięci.



Dzisiaj wiele programów użytkowych dusi się w nierozszerzalnym obszarze pamięci operacyjnej, a światli użytkownicy PC z zazdrością patrzą na Atari ST i Amigę, w których dostępna pamięć jest ograniczona tylko możliwościami adresowymi procesora, a nie decyzjami konstruktorów. Na szczęście w przypadku 386-tki pracującej pod kontrolą systemu Unix lub OS/2 te ograniczenia nie obowiązują.

Co jeszcze można włożyć do środka? Zależy to od zainteresowań i możliwości finansowych właściciela. Dość często kupowaną kartą jest (nie w naszych warunkach)

modem wewnętrzny.

Nazwa sugeruje, że istnieją również modemy zewnętrzne i tak jest w rzeczywistości. Modem służy do podłączenia komputera do telefonu po to, by móc swoje dane i programy przesyłać za pośrednictwem sieci telefonicznej. Istnieje kilka ogólnosiwiatowych sieci komputerowych korzystających wyłącznie z połączeń telefonicznych (najbardziej znana to Fido).

Skaner,

służący do wczytywania zdjęć i rysunków do pamięci komputera, też może opierać się o własną kartę, instalowaną wewnątrz komputera, choć istnieją i takie, które można podłączyć za pośrednictwem łącza szeregowego z karty Multi I/O.

Jakie jeszcze istnieją karty? Każde zadanie, jakie można sobie wyobrazić, daje się rozwiązać dzięki odpowiedniej konstrukcji. Potrzebna jest sieć — nie ma problemu, wystarczy włożyć odpowiednią kartę, żeby móc połączyć komputery ze sobą. Interesuje nas zczytywanie danych doświadczalnych lub sterowanie eksperymentem albo procesem przemysłowym — kupujemy kartę ADDA, zamieniającą sygnały analogowe (A) na cyfrowe (D, jak digital) i z powrotem. Chcemy wczytać obraz z kamery wideo — wystarczy użyć przetwornika wideo, w postaci specjalizowanej karty wkładanej w jedno z gniazd. Potrzebny jest szybszy komputer — mamy do dyspozycji kilka możliwości, od kupienia karty z transputerem, do (jeśli nie o to chodzi) karty SuperCache-286, dzięki której XT zamienia się w AT (oba warianty były opisane w „Bajtku” 12/89). Odpowiednie karty pozwalają na zainstalowanie w XT napędów dysków elastycznych 1.2 i 1.44 MB, normalnie zastrzeżonych dla AT, lub też umożliwiają zrobienie kopii twardego dysku na taśmie wideo w ciągu kilkunastu sekund za pomocą urządzenia zwanego streamer (czyt. strimer). Takie wyliczenie można by jeszcze ciągnąć długo...

IBM — chcąc zdominować rynek mikrokomputerów — wprowadzając swój model PC na rynek opublikował jego bardzo precyzyjny opis techniczny i programowy. Stąd wzięła się wielość producentów, konstruktorów i programów, stanowiących o popularności klonów, które teoretycznie są między sobą kompatybilne, niezależnie od źródła pochodzenia. W praktyce różnie to bywa i przed zakupem nowej karty warto sprawdzić, czy da się ją uruchomić w naszym komputerze — ale to już zupełnie inna historia.

Marcin Borkowski

Prawie każdy wie, że twardy dysk trzeba zaparkować. Po co i na czym to polega, wie kilka procent użytkowników komputerów. Przy założeniu, że każdy kto kupi „Bajtka”, przeczyta ten artykuł, wymieniony wyżej procent wzrośnie z kilku do kilkunastu.



arking strzeżony

Twardy dysk to taka dyskietka z grubej blachy aluminiowej, która kręci się z szybkością 3600 obrotów na minutę. Przy tej szybkości powietrze opływające dysk tworzy na nim cienką poduszkę powietrzną, po której ślizgają się głowice odczytujące i zapisujące informacje na cienkiej warstwie tlenków żelaza pokrywających powierzchnię dysku. Gdyby głowice zetknęły się z tą warstwą i zarysowały ją, zapisana na niej informacja mogłaby ulec nieodwracalnemu uszkodzeniu. Odległość głowic od powierzchni dysku jest mierzona w mikrometrach, ale ryzyko ich zetknięcia jest dzięki poduszce powietrznej znikome — może się to zdarzyć tylko przy bardzo silnych wstrząsach.

Inaczej wygląda sytuacja po wyłączeniu komputera — kiedy dysk przestaje się kręcić, poduszka znika i głowice mogą osiąść na powierzchni dysku. Oczywiście producenci znają ten problem i stosują różne zabezpieczenia, ale doświadczenie wykazuje, że najbezpieczniejszym sposobem na uniknięcie kłopotów jest ustawienie głowic przed wyłączeniem komputera nad nie używanym obszarem dysku — nawet jeżeli zostanie on zarysowany, użytkownik nie traci żadnych danych. To przesuwanie głowic w odpowiednie miejsce nazywa się właśnie parkowaniem.

Pozostaje jeszcze problem wyboru „odpowiedniego” miejsca na zaparkowanie głowic. Z różnych względów najlepszym miejscem jest cylinder znajdujący się tuż za ostatnim cylindrem obsługiwany przez system operacyjny, znajdujący się najbliżej osi dysku. W tym miejscu zetknięcie się głowic z powierzchnią jest niegroźne, gdyż nie ma tam zapisanej żadnej informacji.

Do parkowania dysków służą specjalne programy, najczęściej obdarzone nazwą **park** lub **bye**. Zaparkowanie dysku za ich pomocą polega na wywołaniu programu i ewentualnym udzieleniu odpowiedzi na pytanie, czy rzeczywiście chcemy dysk zaparkować. Dla lubiących



wklepywać i analizować programy assemblerowe przygotowałem króciutki wydruk programu parkującego twardy dysk. Program pobiera parametry twardego dysku, w tym numer ostatniego cylindra zakodowany w rejestrach `ch` i `cl`, zwiększa ten numer o jeden, a następnie żąda przesunięcia głowic. W systemach zawierających więcej niż jeden dysk twardy program nie będzie działał.

Nowsze konstrukcje na ogół zbudowane są tak, że po zaniku napięcia zasilającego samoczynnie przesuwają głowice w bezpieczne miejsce. Dopóki jednak nie mamy stuprocentowej pewności, że tak jest, lepiej nie ryzykować utraty danych i parkować dysk nie zastanawiając się nad tym.

Marcin Borkowski

```
ASSUME cs:code
```

```
code SEGMENT
```

```

org      100h

Start:
    mov    ah,08h      ;Pobranie parametrów
    mov    dl,080h     ;pierwszego twardego
    int     13h        ;dysku.
    jc     error
    cmp    dl,1        ;Czy tylko jeden
    jne    error       ;dysk?
    xchg    cl,ch
    and     ch,0C0h    ;Oblicz numer
    rol     ch,1       ;ostatniego cylindra
    rol     ch,1       ;dodaj do niego jeden,
    xchg    cl,ch      ;i zapisz to
    ror     cl,1       ;w rejestrze cx
    ror     cl,1       ;w formacie znanym
    inc     cl         ;BIOS-owi.
    mov     ah,0Ch     ;Ustaw głowice nad
    mov     dx,080h    ;ostatnim cylindrem
    int     13h
    mov     ah,9
    jc     error
    mov     dx,offset OKinfo
    int     21h

hang:
    jmp     hang

error:
    mov     dx,offset ERinfo
    int     21h
    mov     ax,4C01h
    int     21h

OKinfo:   db 'OK, dysk zaparkowany$'
ERinfo:   db 'Uwaga! Problemy z parkowaniem!'
          db 13,10,'$'

code ENDS

END Start
```


POPRAWKA Z GRAFIKI

Korzystanie z procedur graficznych Turbo Pascala w najprostszym wariancie wymaga spełnienia kilku warunków.

Po pierwsze, program musi znać ścieżkę dostępu do potrzebnego zbioru *.bgi (Borland Graphic Interface), co zwykle oznacza, że zbiór ten jest kopiowany do katalogu, w którym uruchamiamy program. Po drugie, o ile chcemy korzystać z drukarki i klawisza PrintScreen, przed uruchomieniem programu musimy pamiętać o zainstalowaniu **hgcpaint-a** lub **graphics** — w zależności od stosowanej karty graficznej. Po trzecie, w razie nieprawidłowego zakończenia programu, związanego na przykład z błędem obliczeniowym, nie jest automatycznie przywracany tryb tekstowy, co, zwłaszcza w przypadku Herculesa, praktycznie uniemożliwia dalszą pracę. Po czwarte wreszcie, najmniej istotne, procedura uruchamiająca tryb graficzny, **InitGraph**, wymaga przekazania typu karty graficznej i żądanego trybu za pomocą zmiennych, a nie parametrów. Powoduje to konieczność zadeklarowania przynajmniej jednej zmiennej całkowitej, służącej na ogół do jednokrotnego wykorzystania. Dodatkowo, w zestawie dostępnych procedur brak takiej, która pozwalałaby na wydrukowanie zawartości ekranu po jego przygotowaniu pod kontrolą programu, a nie użytkownika. Ma to znaczenie, zwłaszcza gdy chcemy wydrukować kilka wykresów lub ilustracji, których przygotowanie trwa dłużej niż kilkanaście sekund — optymalne rozwiązanie polega na tym, że program sam drukuje ekran po jego przygotowaniu, a nie czeka na naciśnięcie przez użytkownika klawisza PrintScreen.

W celu pozbycia się tych niedogodności napisałem moduł **graph2**, możliwy do wykorzystania niezależnie od stosowanej karty graficznej — do testów użyłem kart Hercules, CGA i EGA, w przypadku innych kart też nie powinno być kłopotów. Do zestawu istniejących procedur przybywają trzy — **StartGraph**, **StopGraph** i **PrintScreen**. Pierwsza z nich służy do uruchamiania trybu graficznego i wymaga podania jako parametru jego numeru — na przykład CGAO, HercMonoHi. W przypadku karty Hercules sprawę można jeszcze bardziej uprościć, rezygnując z podawania parametru. Procedury **StopGraph** i **PrintScreen** działają zgodnie ze swymi nazwami i chyba nie wymagają komentarza.

Do przygotowania modułu do pracy oprócz samego kompilatora Turbo Pascala, potrzebny jest jeszcze program **binobj.exe**, stanowiący część systemu Turbo. Za jego pomocą przekształca się zbiory *.bgi (lub *.com) w zbiory *.obj, z których korzysta linker, czyli program łączący. W naszym przypadku należy wydać polecenie

binobj cga.bgi graph. obj GraphDriver (dla karty CGA). Teraz kompilujemy moduł **graph2** i od tego momentu wszystkie nasze programy, w których z niego skorzystamy, nie będą szukać zbioru *.bgi na dysku. Jeżeli nie macie programu **binobj** lub nie wiecie, z jakiej karty graficznej będziecie korzystać, skasujcie linie zaznaczone trzema gwiazdkami — wtedy wprowadźcie zbiór *.bgi jest niezbędny do uruchomienia programu, ale wszystkie pozostałe udogodnienia są do waszej dyspozycji.

A jakie są te udogodnienia? Zgodne z listą kłopotów, wymienioną na samym początku, oczywiście pod warunkiem korzystania z procedur **StartGraph** i **StopGraph**. (Uwaga — jeżeli z nich korzystacie, nie wolno równocześnie używać **InitGraph** ani **CloseGraph** — w najgorszym wypadku może się to zakończyć zawieszeniem komputera.) Naciśnięcie klawisza PrintScreen spowoduje wysłanie na drukarkę zawartości ekranu niezależnie od trybu graficznego lub tekstowego, przed zakończeniem wykonywania programu zawsze przywracany jest tryb tekstowy, a procedura **PrintScreen** pozwala programowi na wydrukowanie wyników jego pracy niezależnie od Waszej obecności.

Drukarka obsługiwana jest z pominięciem modułu **printer**, dzięki czemu naciśnięcie klawisza **PrintScreen** nie powoduje zawieszenia komputera, co w przeciwnym wypadku mogłoby się zdarzyć. W razie problemów z drukarką spróbujcie zamienić w zmiennej **prnini** sekwencję **#27*'#01** na **#27'L'**, zgodną ze standardem Epson-a.

Marcin Borkowski

```
{ $A+,B-,D-,E-,F-,I-,L-,N-,O-,R-,S-,V- }
unit graph2;
interface
procedure StartGraph(mode : integer);
procedure StopGraph;
procedure PrintScreen; inline($cd/$05);
implementation
uses graph,dos;
var
  savedscreen      : file;
  prn_error,printing : boolean;
  SavedExit,TxtInt5 : pointer;
  screen           : array[0..32767] of byte
                  absolute $B000:$0000;

procedure GraphDriver; external; {***}
{$L GRAPH.OBJ }                  {***}

procedure biosprint(s : string);
var
  rej : registers;
  i   : integer;
begin
  for i:=1 to length(s) do
  begin
    rej.ah:=0; rej.al:=ord(s[i]);
    rej.dx:=0; intr($17,rej);
    if (rej.ah and 41)<>0 then
    begin
      prn_error:=true; EXIT
    end
  end
end;
{$F+ }

procedure _copyscreen;
interrupt;
var
  x,y,iny : word;
  znak,r   : byte;
  rej      : registers;
  prnini   : string;
begin
  if printing then EXIT; printing:=true;
  prn_error:=false; rej.ah:=1;
  rej.dx:=0; intr($17,rej);
  if (rej.ah and 41)<>0 then EXIT;
  biosprint(#27#64#27#51#24#27#108#10);
  x:=GetMaxX+1; y:=0;
  prnini:=#10#13#27*' '#01+chr(Lo(x))+chr(Hi(x));
  while y<GetMaxY-1 do
  begin
    biosprint(prnini);
    for x:=0 to GetMaxX do
    begin
      if prn_error then EXIT;
      for iny:=0 to 7 do
      begin
        znak:=znak shl 1;
        if getpixel(x,y+iny)<>0 then inc(znak)
      end;
      biosprint(char(znak))
    end;
    y:=y+8
  end;
  biosprint(#10#10#13#27'@'); printing:=false
end;

procedure GraphExit;
begin
  closegraph; ExitProc:=SavedExit
end;
{$F- }

procedure StartGraph(mode : integer);
var
  crd : integer;
begin
  crd:=CGA; InitGraph(crd,mode,'');
  SetIntVec(5,@_copyscreen)
end;

procedure StopGraph;
begin
  closegraph; SetIntVec(5,TxtInt5)
end;

begin
  if registerbgidriver(@GraphDriver)<0 then HALT; {***}
  SavedExit:=ExitProc; ExitProc:=@GraphExit;
  printing:=false; GetIntVec(5,TxtInt5)
end.
```


KOMPILATORY

Fortranu

Wiele osób mających kontakt z mikrokomputerem na słowo „FORTRAN” krzywi się niemiłosiernie. Nietrudno zrozumieć taką reakcję, zwłaszcza gdy korzysta się z innych języków — Turbo Pascal, Turbo Basic, Turbo C. Do jakich więc celów może być przydatny język FORMULA TRANslator i co można wycisnąć z jego kompilatorów?

W dziedzinie, którą się zajmuję, istnieją co najmniej dwa poważne zastosowania FORTRANU. Pierwsze — to wykorzystanie napisanych w nim i działających na dużych komputerach programów rozwiązujących skomplikowane zagadnienia chemii czy fizyki kwantowej. Drugie — to wykorzystanie istniejących bibliotek podprogramów numerycznych, które w większości są napisane również w tym języku. Znana mi jest tylko jedna „bogata” dwujęzyczna PASCALO-FORTRANowska biblioteka (dostępna wraz z książką „The Art of Scientific Programming”). Pojęcie „bogatej” biblioteki oznacza, iż zawiera ona ponad 200 procedur (tutaj borlandowska NUMERICA jest raczej kopciuszkiem) o dużej niezawodności działania. W szerokim świecie istnieją zapewne biblioteki numeryczne dla różnych języków, lecz niestety dostępność tych produktów na małym polskim rynku jest znikoma. Bez FORTRANU ani rusz.

Na szczęście istnieje kilka kompilatorów pozwalających na skorzystanie z programów napisanych w tym języku. Spróbujmy im się przyjrzeć. W większości zostały one tak zaprojektowane, aby umożliwić kompilację programów z dużych maszyn liczących. W związku z tym nie ma w nich ograniczenia na rozmiar pojedynczej tabeli do 64 KB — jest on limitowany obszarem dostępnej pamięci podstawowej (640 KB) bądź rozszerzonej (np. do 16 MB w mikrokomputerach 386).

MS-FORTRAN (MicroSoft)

Jest to kompilator generujący zbiory typu OBJ wymagające konsolidacji. Kompilacja przebiega trójfazowo — wykorzystywane są trzy oddzielne programy, z których drugi (PAS2) przetwarza efekt działania pierwszego (PAS1), a trzeci (PAS3) — drugiego. Program fortranowski musi być tak napisany, aby po skompilowaniu segmenty kodu oraz danych nie przekraczały 64 KB każdy. Możliwe jest wykorzystanie koprocatora arytmetycznego oraz praca bez niego. Zaletą programu MS-FORTRAN jest łatwe łączenie procedur skompilowanych przez MS-PASCAL, MS-C oraz MASM. Źródłowe programy fortranowskie muszą być napisane w standardzie F-77. Według mojej oceny MS-FORTRAN jest najpowolniejszym programem tłumaczącym, zarówno pod względem czasu kompilacji, jak i czasu wykonywania programu.

PROFORT (IBM-Ryan-McFarland)

Kompilator zgłasza się komunikatem „Professional Fortran” i chyba ma do takiej autoreklamy prawo. Faza kompilacji wykonywana jest przez jeden program, generujący zbiory OBJ. Możliwe jest zastosowanie debuggera ułatwiającego uruchomienie programu i odszukanie błędów. Liczba danych w programie jest ograniczona rozmiarem pamięci obsługiwanej przez DOS; rozmiar pojedynczej tabeli nie ma ograniczenia do 64 KB, jednak kod wynikowy pojedynczej procedury jak i segmentu głównego nie może przekraczać 64 KB. Niestety, użytkownicy komputerów bez zainstalowanych koprocetorów są pokrzywdzeni — nawet faza kompilacji wymaga obecności „chipów” serii 80X87. Źródłowy program fortranowski musi być utrzymany w standardzie F-77. PROFORT wykonuje automatyczną optymalizację ze względu na czas wykonywania programu (opcję tę można wyłączyć). Pewną niedogodnością jest

struktura procedur assemblerowych, które chcemy wywołać z programu fortranowskiego. W segmencie danych musi być podana nazwa procedury, mimo że występuje ona jako PUBLIC w segmencie kodu.

RMFORT (Ryan-McFarland)

Jest to zmodyfikowana wersja programu PROFORT. Umożliwia pracę w systemie wsadowym jak i interakcyjnym. Kompilacja w obydwu przypadkach wykonywana jest przez ten sam program i trwa jednakowo długo; w obu przypadkach tworzone są zbiory OBJ wymagające konsolidacji oddzielnym programem. Zaletą pracy ze zintegrowanym edytorem jest możliwość ustalania na bieżąco opcji kompilatora oraz kompilacja i konsolidacja wielu zbiorów jedną komendą. RMFORT umożliwia pracę z koprocetorem arytmetycznym lub bez niego (są dwie oddzielne biblioteki procedur i funkcji standardowych) oraz generowanie rozkazów procesora 80286 i koprocetora 80287. Tak jak jego poprzednik — PROFORT, pozwala zastosować tzw. debugger, nie posiada ograniczenia na rozmiar segmentu danych w obrębie pamięci obsługiwanej przez DOS oraz na rozmiar pojedynczej tabeli. Umożliwia, w połączeniu z konsolidatorem PLINK, tworzenie struktur nakładkowych.

WATFOR (University of Waterloo, Canada)

Jest to jedyny znany mi kompilator FORTRANu porównywalny z kompilatorami firmy Borland. Fazy kompilacji i konsolidacji są ze sobą sprzężone i przebiegają bardzo szybko. Możliwe jest bezpośrednie uruchomienie programu w pamięci lub utworzenie zbioru typu EXE na dysku. W tym punkcie kończą się podobieństwa do znanej serii TURBO.

Różnice wizualne i użytkowe są olbrzymie. Nie istnieją rozwijane (pulldown) zestawy opcji. Konfiguracja większości parametrów jest możliwa tylko poprzez oddzielny program konfiguracyjny. Zestaw poleceń służących do współpracy z dyskiem jest ubogi, same zaś nazwy poleceń nietypowe (PUT zamiast SAVE, GET zamiast READ itp.). Najbardziej uciążliwe jest wyłączenie z działania klawisza RETURN! Należy stosować klawisz F5 lub inny ustalony np. w programie KEYMAP. Niestety program KEYMAP nie umożliwia zamiany F5 na RETURN. Inną niedogodnością jest istnienie dwóch kompilatorów — jednego dla systemu bez koprocetora arytmetycznego i drugiego dla systemu z koprocetorem. Tyle o wadach.

Do niewątpliwych zalet zaliczyć trzeba możliwość korzystania z bibliotek skompilowanych procedur (tak jak poprzednio omawiane kompilatory), włączania do programu zbiorów typu OBJ oraz źródłowych procedur fortranowskich. Możliwa jest także współpraca z procedurami skompilowanymi przez PROFORT i RMFORT. Dla pewnej grupy tradycjonalistów zaletą będzie możliwość pracy w trybie wsadowym. W porównaniu z opisanymi kompilatorami wsadowymi bardzo wygodna jest diagnostyka błędów. Program WATFOR pozwala wykorzystać szereg rozszerzeń do standardu F-77 (WHILE, LOOP, SELECT CASE itp. wraz z żądaniem zadeklarowania każdej zmiennej), czyniących z niego język PASCALO-podobny.

FL-FORTRAN (MicroSoft)

Kompilator FL jest następcą, niezwykle rozbudowanym, serii kompilatorów MS. Kompilacja odbywa się wsadowo, tworzone są zbiory OBJ wymagające konsolidacji. W konsolidacji programu wykorzystywanych jest pięć bibliotek o dużych rozmiarach (ponad 100 KB każda), co czyni ten proces niezwykle czasochłonnym. Do dużych zalet tak tradycyjnie ujętego kompilatora należą: możliwość włączania w kod źródłowy programu procedur napisanych w języku C, praca bez lub z koprocetorem arytmetycznym, wykorzystanie procesora 80286 wraz z 80287 oraz zdolność do tworzenia struktur nakładkowych. Program fortranowski może zawierać rozszerzenia w stosunku do standardu F-77, lecz nie tak szerokie jak w programie WATFOR. Ograniczeniem dla rozmiarów segmentów danych i kodu jest rozmiar pamięci obsługiwanej przez DOS.

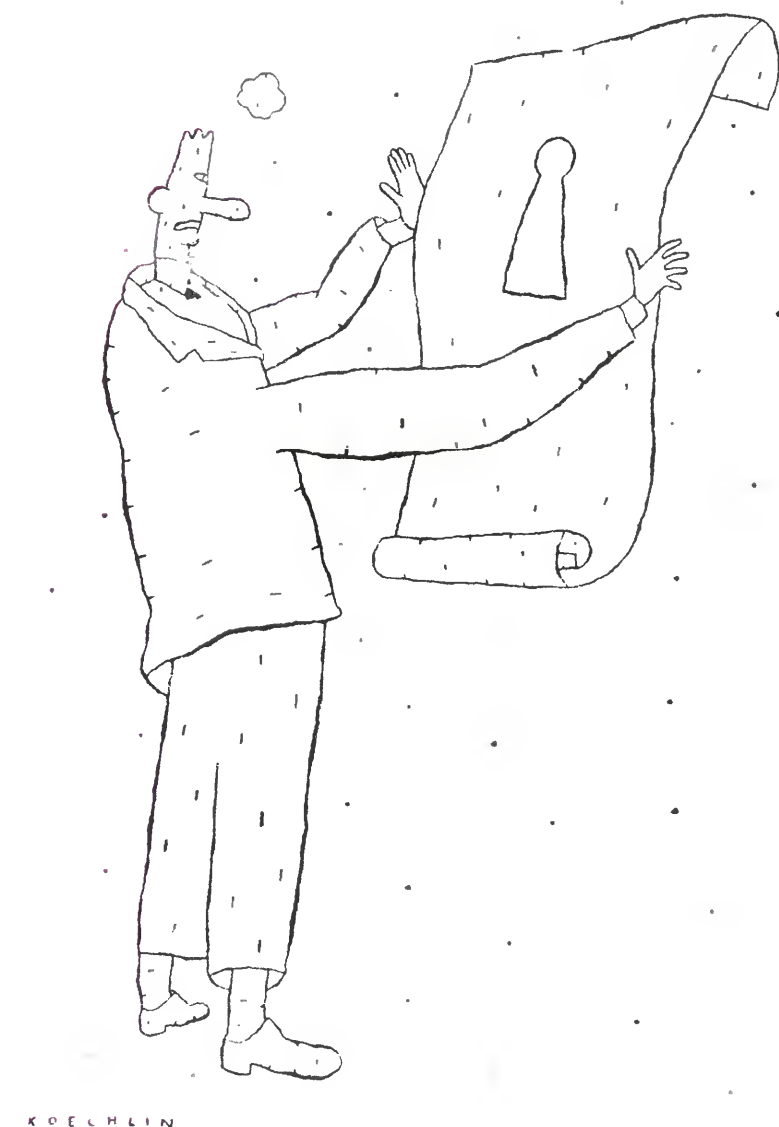
NDP i SVS-FORTRAN (Phar Lap)

Dwa wymienione razem kompilatory potrafią wykorzystać szersze możliwości procesora 80386 i są przewidziane do pracy w systemie z taką jednostką centralną. Umożliwiają tworzenie programów o rozmiarach kodu wynikowego przekraczającego 640 kB. Oba pozwalają na pracę w systemie zawierającym koprocetor arytmetyczny 80287, 80387 lub Weitek oraz, w najgorszym wypadku, na pracę bez koprocetora. Są to kompilatory wsadowe generujące zbiory typu OBJ przeznaczone do konsolidacji przez wyspecjalizowany program 386LINK. Efektem końcowym może być zbiór typu EXE bądź EXP. Zbiór EXP wykonywany jest pod kontrolą specjalnego programu RUN386 i może wykorzystać całą dostępną pamięć. Program RUN386 nie może być uruchamiany z poziomu XTREE lub NORTON COMMANDER i ostrzega przed skasowaniem RAM-dysku umieszczonego w rozszerzonej pamięci. Dopelnieniem systemu 32-bitowego FORTRANu jest 386ASM — assembler 32-bitowy. Dodać należy, że program źródłowy musi spełniać wymogi standardu F-77.

W mojej opinii zastosowanie języka FORTRAN dla mikrokomputerów IBM-PC ogranicza się do implementacji programów obliczeniowych przenoszonych z dużych maszyn. Co prawda, zetknąłem się z profesjonalnym programem graficznym napisanym w tym języku, lecz szybkość jego działania pozostawiała co nieco do życzenia. Jeżeli o grafice słowo, to każdy program fortranowski może wykorzystywać procedury graficzne zawarte w bibliotece GKS (Graphics Kernel System).

Zaprezentowane krótkie zestawienie znanych mi kompilatorów języka FORTRAN jest jedynie zasygnalizowaniem tematu. Ciekawe, że Borland, znany ze świetnych kompilatorów, do tej pory nie zajął się tym językiem. Zapewne jest to związane z polityką firmy, ale przydałby się Turbo Fortran...

Tomasz Grochowski



DESK — LINK

— oprogramowanie sieciowe bez karty

Jeśli chcemy połączyć ze sobą nie więcej niż dwa komputery, to zakup specjalnej karty sieciowej mija się często z celem.

W takim wypadku rozwiązaniem z wyboru staje się zwykle tzw. Zero-Slot Net, czyli oprogramowanie, które realizuje większość operacji sieciowych, bez konieczności nabywania dość drogiej karty.

W kwietniowym numerze PC Magazine z roku 1990 w teście porównawczym niezwykle wysoko oceniono program **Desk-Link** firmy **Travelling Software Inc.** Pozwala on na uwspólnienie zasobów dwóch komputerów połączonych ze sobą przez złącze szeregowo RS 232C. Dotyczy to stacji dysków i drukarek obsługiwanych przez interfejsy równoległe typu Centronics. Dodatkowa możliwość to prosty substytut poczty elektronicznej, tzn. wymiana informacji tekstowej przy pomocy klawiatury i ekranu.

Program pracuje w tle umożliwiając indywidualne wykorzystanie każdego z dwóch komputerów. Obce napędy widziane są jako kolejne stacje dysków. Poza operacjami Diskcopy, Format lub podobnymi możliwe jest korzystanie z nich jak z własnych napędów. Jest to szczególnie użyteczne przy kopiowaniu plików z dyskietki 3.5" w laptopie na dyskietkę 5.25 w dużym komputerze. Drugim cennym

zastosowaniem jest wymiana danych i programów między dyskami twardymi dwóch komputerów. Dzięki dużej szybkości (115200 bodów, a efektywnie 77 280) operacja taka zajmuje mniej czasu niż wymiana plików na dyskietkach.

Bardzo użyteczna jest możliwość korzystania z obcej drukarki. W tym wypadku nie ma żadnych ograniczeń: możemy drukować kopię ekranu, swój tekst z edytora lub cokolwiek innego. Co najwyżej trochę spowolnimy drugi komputer, do którego podłączona jest drukarka. Ponieważ operacje te są buforowane, to często dla osoby pracującej na komputerze bez drukarki jest to szybsze i wygodniejsze niż przy pracy z drukarką bezpośrednio do niego przyłączonej.

Niektóre programy do wymiany informacji między komputerami używają także złącza równoległego. Desk-Link nie daje takiej możliwości, ale niewiele ustępuje konkurentom, a w swojej klasie, programów pracujących przez złącze RS 232C, jest najszybszy.

Na rys. 1a i 1b przedstawiono dwa przykładowe ekrany menu konfiguracyjnego. Pierwszy dotyczy laptopa wyposażonego w stację dysków 3.5", drugi — komputera wyposażonego w dwie stacje 5.25" (1.2MB i 360 KB), dysk twardy, ramdysk i drukarkę podłączoną do portu równoległego LPT1.

Program Desk-Link ma wiele zalet i nieliczne wady. Do tych ostatnich należy m.in. niezgodność oprogramowa-

Expanded Options Menu, version 4.01 (c) 1989 Traveling Software, Inc.
DESK-LINK

Printer			System Selections	
Printer:	Local	<u>Remote-LPT1</u>	Starting Drive:	D:
Description:	Bondwell Laptop		Number of Drives:	4
Characters in Spool Buffer:	0		Print Buffer Size:	16k
Remote Drive Reference Table			Hotkeys	
Local Drive	Remote Drive	Use <F10> to Edit Description	Menu Hotkey:	ALT-D
-----			Talk Hotkey:	ALT-T
D: ---->	A:	FDD 5.25 1.2 MB	Talk Box:	<u>Enabled</u> Disabled
E: ---->	B:	FDD 5.25 360 KB	Serial Port Parameters	
F: ---->	C:	HDD 20 MB	Serial Port:	<u>COM1</u> COM2 COM3 COM4
G: ---->	D:	Remote Randisc	Baud:	115200 57600 38400 19200 9600

Help Save Restore Empty-spool Quit
DESK-LINK Active

Rys. 1a Menu konfiguracyjne (laptop)

Expanded Options Menu, version 4.01 (c) 1989 Traveling Software, Inc.
DESK-LINK

Printer			System Selections	
Printer:	Local	Remote-LPT1	Starting Drive:	E:
Description:	Future XT (Host)		Number of Drives:	2
Characters in Spool Buffer:	0		Print Buffer Size:	16k
Remote Drive Reference Table			Hotkeys	
Local Drive	Remote Drive	Remote Description	Menu Hotkey:	ALT-D
E: ---->	A:	FDD 3.5" 720 KB	Talk Hotkey:	ALT-T
F: ---->	C:	Remote Ramdisc	Talk Box:	<u>Enabled</u> Disabled
			Serial Port Parameters	
			Serial Port:	COM1 <u>COM2</u> COM3 COM4
			Baud:	<u>115200</u> 57600 38400 19200 9600

Help Save Restore Empty-spool Quit
DESK-LINK Active

Rys. 1b Menu konfiguracyjne (komputer XT)

nia z procedurami standardowego modułu sieciowego NetBIOS oraz brak możliwości pracy z dyskami CD-ROM.

Olbrzymia użyteczność opisywanego programu, wygoda i łatwość obsłu-

gi oraz doskonałe parametry techniczne są przyczyną, dla której łatwiej zrozumieć dość wysoką cenę tego produktu — 169.95 USD.

Jonasz Mayer

POLSKIE LITERY

Kiedy pojawiły się w Polsce pierwsze komputery osobiste, a było to ładnych kilka lat temu, ministerstwa i urzędy miały na głowie Znacznie Ważniejsze Sprawy, więc specjalnie nie mieszały się do tej dziedziny.

Komputerom wyszło to na dobre, z jednym wyjątkiem. Są nim polskie litery. Zabrakło gdzieś na górze jednego faceta, który pomyślałby przez chwilę, i wydał polską normę opisującą przyporządkowanie liter kodom ASCII. Skoro nie zrobił tego nikt na górze, zajęli się tym ludzie na dole. Każda niemal firma mająca do swej dyspozycji elektronika i programator EPROM-ów lansowała własny standard, sugerując, że jest on najlepszy, i na pewno przetrzyma standardy wszystkich innych firm.

Firmy i niektórzy urzędnicy zniknęli, bałagan i brak normy pozostały. Ostatnio wprowadzie na szczeblach ministerialnych mówi się o konieczności wprowadzenia jakiegoś standardu, ale że o prawnej ochronie oprogramowania też się mówi, od wielu lat i na wielu szczeblach, nie należy się raczej spodziewać szybkiego efektu.

A o których standardach się „mówi”? Najczęściej o dwóch. Za jednym, moim i nie tylko moim zdaniem najgorszym, stoi IBM — jest to standard Latin-2. Doświadczenie uczy, że standardy lansowane przez tę firmę są złe, i przyjmują się błyskawicznie. Drugi standard to Mazovia. Oprócz nich istnieje jeszcze jeden, dość popularny w Polsce — DHN. Na wszelki wypadek, aby umożliwić naszym czytelnikom wyrobienie

	Mazovia	Latin-2	DHN
ą	(134) ă	(165) ã	(137) ë
ć	(141) ĩ	(134) à	(138) è
ę	(145) æ	(169) ħ	(139) ï
ł	(146) Æ	(136) ê	(140) î
ń	(164) ñ	(228) Σ	(141) ï
ó	(162) ó	(162) ó	(142) Å
ś	(158) Ŗ	(152) γ	(143) Å
z	(167) ų	(190) ħ	(145) æ
ż	(166) ă	(171) ¼	(144) é
Ą	(143) Å	(164) ñ	(128) Ç
Ć	(149) ð	(143) Å	(129) ü
Ę	(144) é	(168) ħ	(130) é
Ł	(156) £	(157) ¥	(131) ä
N	(165) ñ	(227) π	(132) ä
Ń	(163) ú	(224) α	(133) à
Ś	(152) γ	(151) ù	(134) à
Z	(161) ĩ	(189) ı	(136) ê
Ż	(160) à	(141) ĩ	(135) ç

sobie zdania na ten temat, drukujemy dzisiaj wszystkie trzy standardy. Najlepsze wrażenie robi standard Mazovii, ale czy te głupie kilkadziesiąt czy kilkaset tysięcy jego użytkowników może być poważnie brane przez bądź co bądź poważnych decydentów?

Marcin Borkowski



MAM DRUKARKĘ I ...

...po kilkudziesięciu lub kilkuset stronach, w zależności od typu drukarki i trybu drukowania, jakość druku wyraźnie się obniża. Jeszcze kilkanaście wydrukowanych stron i prawie nic nie widać. Jest to moment, w którym dla wielu użytkowników kwestia kasety z taśmą barwiącą jawi się jako problem. Problem, gdyż w odróżnieniu od krajów na zachód od Odry, brak w Polsce sklepów oferujących kasety wszystkich typów z ostatnich lat. Ponadto nabycie kasety, jeżeli jest do kupienia, wiąże się ze znacznym wydatkiem.

Instytucje handlowe często lekceważą klientów, jak np. Centralna Składnica Harcerska oferująca swego czasu drukarkę Commodore MPS 803, już od ponad roku nie pamięta o swoich dawnych klientach i nie sprowadza kasety do niej. Nieliczne sklepy komisowe i coraz liczniejsza rzesza pośredników oferują kasety do drukarek najbardziej popularnych, „chodliwych” typów, mając zapewne na względzie bardziej zyski niż interesy użytkowników, zwłaszcza starszych typów drukarek. Jednocześnie większość proponowanych na naszym rynku kasety, pochodzenia zachodnioeuropejskiego i dalekowschodniego, znacznie odbiega jakością od produktów oryginalnych, sygnowanych przez firmy produkujące drukarki. Te ostatnie nie są obiektem zainteresowania importerów prywatnych. Szukają oni towaru najtańszego, a jego zachodnie pochodzenie niemal wyklucza pytanie o jakość, sugerując produkt najwyższej lub bardzo dobrej jakości. Zdarza się również, że po wyjęciu kasety z fabrycznego opakowania uzyskujemy bardzo słaby wydruk, co świadczy o tym, że jest przeterminowana. Należy zwrócić uwagę, że w naszych warunkach ceny tych kasety nie są znacząco niższe niż kasety oryginalnych. Wyższa cena tych ostatnich jest z nawiązką rekompensowana ich jakością.

Co składa się na tę jakość? Jednym z czynników jest jakość samej nylonowej taśmy barwiącej. Najważniejszym wyróżnikiem jakości taśmy jest jej zdolność wchłaniania tu-

szu. Wyższość taśm oryginalnych w tym względzie widać szczególnie wyraźnie w procesie regeneracji taśmy (o regeneracji obszerniej w dalszej części artykułu). Chłoną one znacznie więcej tuszu niż taśmy innych producentów.

Innym czynnikiem, widocznym po kilku procesach regeneracji, jest jakość połączenia taśmy — najczęściej taśma w kasie jest łączona w dwustronną wstęgę bez końca lub we wstęgę Möbiusa. Bardzo dobrą jakością wyróżniają się oryginalne kasety firmy STAR, gdzie uszkodzenia na połączeniu pojawiają się po 3—4 procesach regeneracji, w odróżnieniu od kasety innych producentów, gdzie uszkodzenia zdarzają się już pod koniec zerowego przebiegu, a są regułą po 1—2 regeneracjach.

Kolejnym, równie istotnym czynnikiem korzystnie wyróżniającym kasety oryginalne jest jakość mechaniczna. Nie wdając się w szczegóły różnorodnych rozwiązań technicznych, jakie tu można wyliczyć, należy jednoznacznie stwierdzić, że ilość zacięć taśmy wewnątrz kasety, przekręcania się taśmy z pojawieniem się tego przekręcenia na zewnątrz kasety lub trwałych zagięć taśmy poprzez złe ułożenie jej w kasie — w kasetach firmy STAR w porównaniu z kasetami innych producentów ma się jak 1 do 100.

Wydaje się uzasadniony wniosek, że korzystanie z nieoryginalnych kasety z taśmą barwiącą jest rozrzutnością.

Warta odnotowania jest jeszcze informacja, że spotyka się kasety nieznanego z nazwy producenta zachodniego do Amstrada PCW, gdzie taśma jest trzykrotnie krótsza niż w kasie oryginalnej. Naturalnie żywotność takiej kasety jest trzykrotnie mniejsza, czego nabywający kasety — z oczywistych względów — nie może nawet podejrzewać.

Znajdujący się w kłopotach użytkownik drukarki, wobec wyeksploatowania kasety z taśmą barwiącą, ma możliwość oddania jej do regeneracji do zakładu usługowego. Niestety jakość tych usług jest zróżnicowana. Niektóre z zakładów paskudzą taśmę barwiącą jakąś substancją, mającą niewiele wspólnego z *wysoko specjalistycznym* tuszem do barwienia taśm (technologia wytwarzania tuszu stanowi tajemnicę producenta, a sam tusz nie jest artykułem powszechnie dostępnym na rynku). Substancja ta po wyschnięciu impregnuje taśmę suchą pozostałością, pozbawiając ją chłonności, i pokrywa ją czymś w rodzaju sztywnej skorupy. Skutki takiej działalności „usługowej” są fatalne zarówno dla drukarki, a zwłaszcza jej głowicy, jak i użytkownika. Wydruk uzyskany z takiej taśmy ma nierównomierne nasycenie, żywotność taśmy jest bardzo krótka, a co gorsza — tak „zregenerowana” nadaje się do wyrzucenia.

Prawidłowe wykonanie zabiegu regeneracyjnego przez wyspecjalizowany zakład usługowy przy użyciu właściwego tuszu daje jednak dobre rezultaty. Żywotność takiej taśmy nie odbiega od taśmy nowej, a zabieg regeneracji można powtarzać kilkakrotnie. Ocenę taką potwierdza test

przeprowadzony przez miesięcznik „Mikroklan” (nr 10/87 i nr 5/88).

Kłopoty użytkowników związane z eksploatacją drukarek komputerowych wywołują u niektórych z nich instynkt majsterkowicza — niestety, często zabójczy, zwłaszcza dla taśm barwiących i głowicy drukarek. Realizuje się on w nasączeniu taśm barwiących domowym sposobem przy użyciu tuszu do stempli zwykłych lub metalowych i zszywaniu zerwanych taśm nitką. Stanowczo odradzam stosowanie takich praktyk i proponuję majsterkowanie w innych dziedzinach.

Na prawidłową i efektywną eksploatację kasety z taśmą barwiącą ma również wpływ właściwe operowanie dociskiem głowicy drukarki. Wśród użytkowników spotyka się takich, którzy nie wiedzą o istnieniu tego mechanizmu i nie znają lokalizacji jego dźwigni (head gap lever). Pozytywnie należy dostosowywać do grubości używanego papieru i stanu nasycenia tuszem taśmą barwiącą. Prawidłowe ustawienie głowicy osiągnąć można przez wykonanie krótkiego wydruku testowego, rozpoczynając test od położenia głowicy w pozycji maksymalnie odchylonej od wałka. W przypadku złej jakości druku należy test powtarzać, przesuwając każdorazowo głowicę z minimalnym skokiem w kierunku wałka, aż do uzyskania pożądanej jakości.

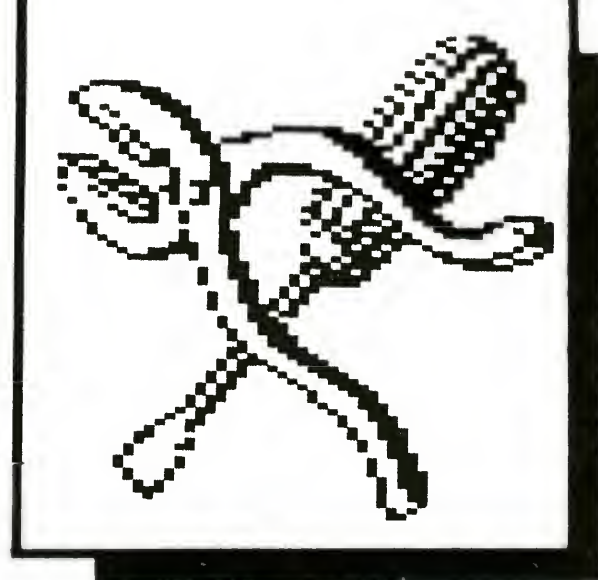
Należy również pamiętać o okresowym przetarciu czystą, miękką szmatką (flanelą) głowicy w miejscu, gdzie styka się ona z taśmą barwiącą, w celu usunięcia gromadzącego się tam kurzu i drobinek nylonu nasycenych tuszem.

Innym ważnym zabiegiem pielęgnacyjnym jest usuwanie gromadzących się wewnątrz drukarki papierowych krążków pochodzących z bocznej perforacji papieru, a także osiadającego tam kurzu. Dobrym sposobem na usunięcie kółeczek może być wysypanie ich poprzez odwrócenie drukarki do góry dnem po uprzednim zdjęciu pokrywy drukarki i przedmuchiwanie wnętrza, używając w tym celu np. odkurzacza jako dmuchawy przez przełożenie węża z otworu ssącego do otworu dmuchającego. Czynności odwracania i wydmuchiwania należy powtórzyć trzykrotnie.

Michał Daszewski

OD REDAKCJI: Autor artykułu prowadzi ponad 3,5 roku zakład usługowy specjalizujący się w regenerowaniu taśm barwiących drukarek komputerowych. Nasza redakcja od początku 1988 roku korzysta z usług tego zakładu.

Proponujemy, w interesie użytkowników drukarek, przysyłanie do redakcji opinii na temat jakości usług zakładów usługowych oferujących regenerację taśm barwiących. Opinie te zostaną zamieszczone w naszym piśmie.



Układ Watch — Dog

Wyobraźmy sobie mikrokomputer sterujący domową pralką. Po załadowaniu pralki bielizną i uruchomieniu całości, komputer otwiera zawór dopływowy. Pralka napełnia się wodą. Napełnianie trwa tak długo, aż komputer otrzyma sygnał z czujnika poziomu wody, że pralka jest pełna.

Nietrudno jest przewidzieć skutki awarii komputera w trakcie nalewania wody. Nagła powódź w mieszkaniu może zniechęcić do używania komputerów praktycznie każdego.

Czy awaria komputera jest aż tak bardzo prawdopodobna? Okazuje się w praktyce, że typowa awaria sprzętowa sterownika zdarza się niezwykle rzadko. Znacznie częściej, w trakcie realizacji programu obsługi urządzenia mikrokomputer potrafi się „zawiesić”. Przyczyn tego zjawiska może być bardzo dużo, wymienię tylko kilka podstawowych: zakłócenia, krótkotrwałe spadki napięcia zasilania, błędy w oprogramowaniu, prawa Murphy'ego itp.

Projektanci niezawodnych sterowników mikrokomputerowych muszą się zabezpieczać przed takimi awariami. Najpopularniejszym układem zabezpieczenia są układy Watch — Dog. Tym zabawnym

terminem określa się układy kontrolujące pracę procesora. Układ taki po wykryciu zawieszenia się mikrokomputera, generuje sygnał przerwania lub reset i przywraca poprawną pracę. Reakcją sterownika na te sygnały powinno być sprawdzenie stanu obiektu i zlikwidowanie zagrożenia. Można zatem uniknąć stanu trwałej utraty kontroli nad urządzeniem.

Jak się buduje takie układy? Jedno z najprostszych rozwiązań jest przedstawione na rysunku 1. Działanie układu jest proste. Każda operacja zapisu lub odczytu z pamięci układu sterownika powoduje zwiększenie zawartości licznika o jeden. Operacje odczytu i zapisu z pamięci są dokonywane przez sterownik zawsze, niezależnie od tego czy pracuje on poprawnie, czy też zawiesił się i wykonuje nieskończoną pętlę złożoną z kilku instrukcji. Jeśli licznik się przepełni, to na jego wyjściu pojawia się impuls który wywołuje przerwanie lub resetuje komputer. Do zadań programisty należy takie skonstruowanie programu, aby licznik był okresowo kasowany. W trakcie normalnej pracy nigdy nie może dojść do jego przepełnienia. W programie obsługi urządzenia między fragmentami kodu należy więc umieścić instrukcje kasujące licznik.

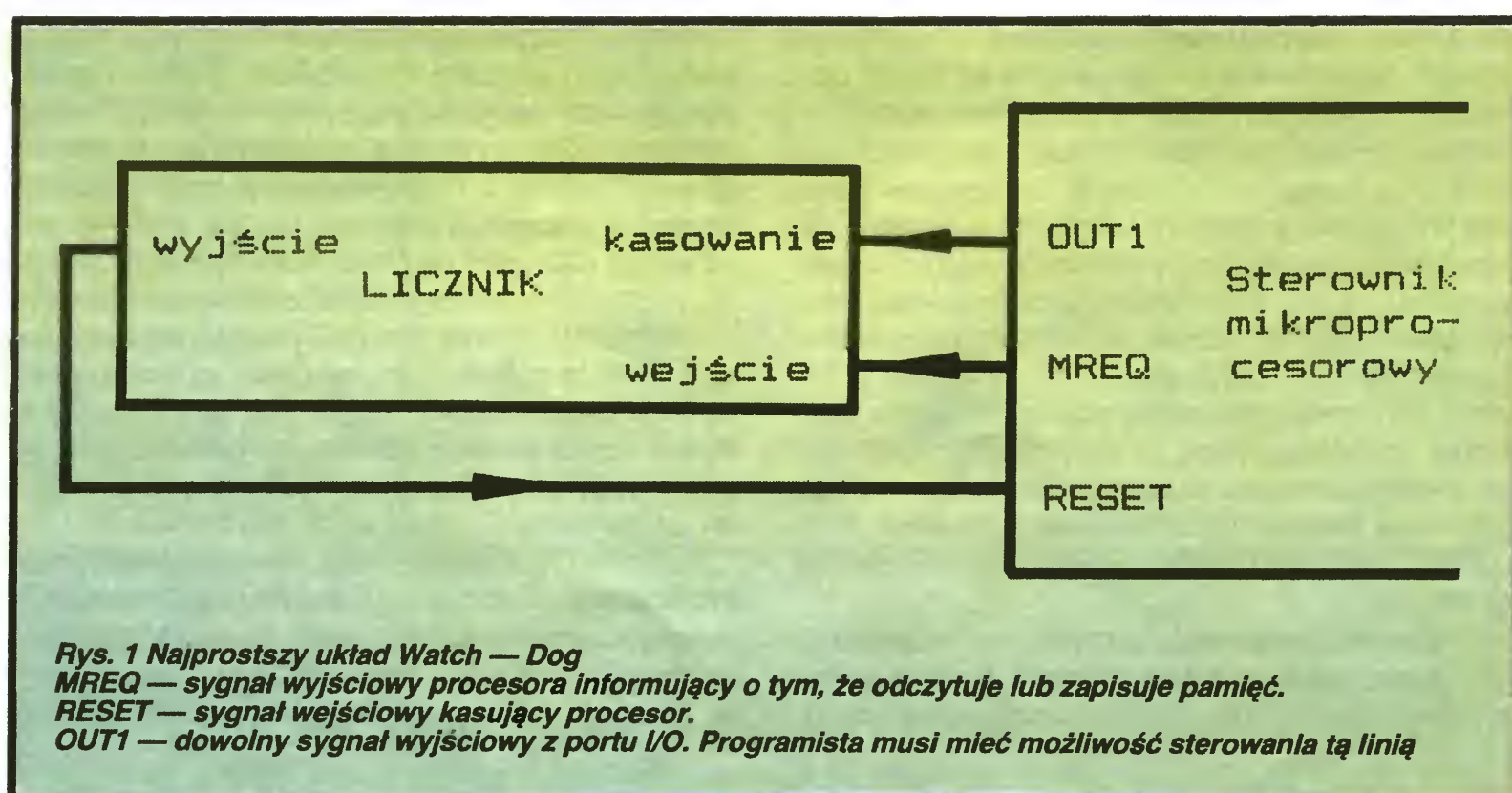
Jeśli komputer się zawiesi, to licznik nie będzie kasowany i po krótkiej chwili nastąpi



przerwanie. Oczywiście w praktycznych układach układ kasowania jest bardziej rozbudowany, tak, aby zawieszenie komputera bezpośrednio na jednej instrukcji do kasowania licznika nie spowodowało braku działania układu. Typowo, aby skasować licznik należy wykonać w ściśle określonej kolejności kilka instrukcji umieszczonych w różnych miejscach programu.

A może ktoś z Czytelników pokusi się o opracowanie układu Watch — Dog do Atari?

Robert Magdziak



„Okna na świat” — ciąg dalszy

MIERNIK POJEMNOŚCI

W poprzednim numerze opisano uniwersalny interfejs z układem 8255. W tym artykule pokażemy, jak przy pomocy tego interfejsu i 3 dodatkowych elementów można mierzyć pojemność kondensatorów.

Schemat ideowy miernika jest na rys. 1. Jak widać, nie jest zbyt skomplikowany. Podstawą jest popularny timer 555 produkowany w Polsce pod oznaczeniem ULY 7855, pracujący w układzie wyzwalanego przerzutnika monostabilnego. Podanie ujemnego impulsu na nogę 2 układu 555 wyzwała ten układ powodując powstanie dodatkowego impulsu na wyjściu (noga 3) o czasie trwania T_i określonym wartościami R_w i C_x . R_w toopornik wzorcowy, a C_x to mierzona pojemność. Ponieważ czas impulsu T_i dla układu 555

$$T_i = 1.1 \cdot R_w \cdot C_x$$

a R_w jest stały więc ostatecznie

$$T_i = k \cdot C_x$$

$$C_x = T_i / k$$

Z ostatniego wzoru widać, że mierzona pojemność C_x jest wprost proporcjonalna do czasu T_i . Mierzając ten czas przy pomocy komputera i dzieląc go przez odpowiedni współczynnik „k” można pośrednio mierzyć pojemność.

Wejście wyzwalające układu 555 jest połączone z bitem zerowym portu PA. Wyjście tego układu jest połączone z bitem zerowym portu PC. Port PA pracuje jako wyjściowy, port PC jako wejściowy.

Zmiana stanu bitu zerowego portu PA z wysokiego na niski wyzwała układ 555, który generuje dodatni impuls o czasie zależnym od wartości opornika R_w i mierzonej pojemności C_x . Ponieważ noga 3 układu 555 jest połączona z bitem zerowym portu PC więc testując ten bit w programowej pętli można łatwo zmierzyć czas trwania impulsu.

Program obsługujący miernik składa się z dwóch części. Część w Basicu (listing 1) wyświetla wynik pomiaru na ekranie. Część druga (listing 2) ze względu na szybkość napisana w asemblerze — inicjalizuje port i zlicza czas trwania impulsu T_i .

Inicjalizacja zrealizowana jest w liniach od 120 do 160. Słowo sterujące STER wysłane do rejestru sterującego RS ustawia port PA interfejsu jako wyjściowy, a port PC jako wejściowy. W liniach 220—240 wysyłany jest ujemny impuls wyzwalający układ 555. Linie 250—280 to programowy licznik czasu. Na początku inkrementowany jest licznik HL, potem testowany zerowy bit portu PC i jeśli równy jest 1 (to znaczy, że trwa impuls T_i) następuje kolejna inkrementacja. W przeciwnym wypadku wartość rejestru HL zostaje zapisana w zarezerwowanym miejscu pamięci o symbolicznym adresie „ile” (ile pętli) i następuje powrót do BASIC-a.

Dla wartości opornika $R_w = 9.1 \text{ MOhm}$ zakres pomiarowy wynosi 750 nF. Zmienna „wsp” w listingu 1 to znany nam z wzorów na czas T_i współczynnik „k”. Zmienną „zer” zerujemy wskazania miernika bez dołączonego kondensatora C_x (stałe pojemności montażowe i pojemność przewodów). W linii 110 wyświetlana jest ilość obiegów pętli zliczania czasu. Dla zakresu 750 nF 1 obieg odpowiada ok. 12 pF. Daje to wskazówkę co do dokładności wykonywanego pomiaru (rośnie wraz ze wzrostem mierzonej pojemności).

STROJENIE MIERNIKA

Zmieniając wartość zmiennej „zer” w zamieszczonym obok programie zerujemy wskazania miernika bez dołączonego C_x . Teraz dołączamy dokładny kondensator o wartości 100—500 nF i zmieniając zmienną „wsp” staramy się uzyskać prawidłowe wskazania. Dokładność pomiarów zależy od stabilności opornika R_x oraz stabilności napięcia zasilania.

Opisany miernik niekoniecznie musi służyć do pomiaru pojemności kondensatorów. W technice pomiarowej często wykorzystuje się pomiar pośredni. Np. można mierzyć zawartość tłuszczu w maśle mierząc jego pojemność. Nieznacznie modyfikując opisany miernik z układem 555, można mierzyć opór, natężenie oświetlenia lub temperaturę. Jeśli będzie zainteresowanie to w przyszłych numerach opiszemy te modyfikacje i przedstawimy przykładowe programy.

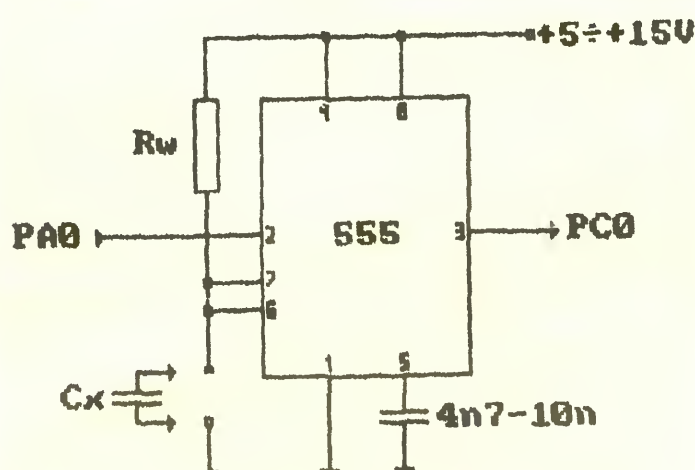
Grzegorz Bujanowski

LISTING 1

```
20 REM Miernik pojemności
30 REM Zakres do 750 nF
40 REM @ RGB
50 REM Rw=9.1 MOhm
60 PRINT AT 21,28;"@RGB": PRIN
T AT 2,10;"Miernik C"
70 LET wsp=.86249
80 LET zer=1
90 RANDOMIZE USR 65007: PRINT
AT 8,10;"
95 PRINT AT 8,10:INT (((PEEK 6
5033+256*PEEK 65034)-zer)/wsp)/1
00
100 PRINT AT 8,17;"nF"
110 PRINT AT 17,10;" : P
RINT AT 17,10;PEEK 65033+256*PEE
K 65034
120 PRINT AT 17,17;"cykli pomia
r"
130 PAUSE 20
140 GO TO 90
150 CLEAR 64999: LOAD ""CODE 65
000: RANDOMIZE USR 65000
160 GO TO 60
170 SAVE "Mier. C.3" LINE 150:
SAVE "mier. mc"CODE 65000,40
```

LISTING 2

```
10 ;Miernik pojemności
20 ;
30 ; RGB
40 ;
50 ; ORG 65000
60 ;
70 PORT_A EQU 31
80 PORT_C EQU 159
90 RS EQU 223
100 STER EQU %10001001
110 ;
120 INIT PUSH AF
130 LD A,STER
140 OUT (RS),A
150 POP AF
160 RET
170 ;
180 POCZ PUSH AF
190 PUSH HL
200 LD HL,00
210 LD A,0
220 OUT (PORT_A),A
230 LD A,1
240 OUT (PORT_A),A
250 PETLA INC HL
260 IN A,(PORT_C)
270 BIT 0,A
280 JR NZ,PETLA
290 LD (ILE),HL
300 POP HL
310 POP AF
320 RET
330 ;
340 ILE DEFW 0
```



KAŻDE
PAŃSTWA ZAMÓWIENIE
ZREALIZUJEMY



LHK Electronics Service S.A
81-736 Sopot
ul. Architektów 1A
* tlx.512742 BSP PL
PC/XT/AT/386/LAP

w dowolnych konfiguracjach, po atrakcyjnych cenach proponujemy: HD 20,30,40,80 MB, monitors 14":amber, paper-white, EGA Bogaty wybór akcesoriów komputerowych:

- a) filtry
- b) obudowy
- c) ruchome ramiona
- d) dyskiety itp.

LHK

TWOIM TRAFNYM
WYBOREM

Uwaga! Odbiorcom hurtowym
udzielamy do 12% rabatu

NOWOŚĆ!



DYSKIETKA CZYSZCZĄCA NAPĘDY
GŁOWIC KOMPUTEROWYCH

Dyskietka czyszcząca zapobiega zużyciu się głowicy, przedłuża trwałość komputera, eliminuje wszelkie zanieczyszczenia bez ryzyka uszkodzenia komputera. Dyskietka może być używana wielokrotnie, posiada wymienne wkłady i wystarcza na kilka miesięcy pracy komputera. Cena promocyjna 25000 zł.

Zamówienie

zamawiam sztuk * 25000 zł
Nazwisko/firma
adres
wysyłka za zaliczeniem pocztowym

LHK prowadzi regenerację taśm do wszystkich typów drukarek
Regenerując taśmę — oszczędzasz pieniądze
Usługę realizujemy w 48 godzin —
wysyłka za zaliczeniem pocztowym.



W

ESTERN W POLSKIM STYLU

Jeniec stał trzymany przez dwóch rosnących wojowników i wpatrywał się przymrużonymi oczami w mieniący się bladozielonkawą poświatą totem nieprzyjacielskiego plemienia. Sąd dawno już się skończył: wszyscy oczekiwali, aż starszyzna wyda wyrok.

— Przywiązać go do joysticka męczarni! — powiedział wielki wódz. Całe plemię jak jeden mąż zawyło z triumfem...

Nie pierwszy to raz w kraju Pol-skah dochodziło do krwawych rozpraw pomiędzy zwaśnionymi plemionami. Setki takich dramatów rozgrywały się już wcześniej, lecz nie na tak wielką skalę; oblicza się, że w waśniach bierze obecnie udział około 2 milionów wojowników. Podłoże tych walk nigdy nie będzie dobrze znane przyszłym historykom — chodzi nie o terytoria, urazy osobiste czy kopalnie złota, lecz o totem i wyrzuty na nim napis. Za ten napis i sam kształt totemu wielu mężnych wojowników oddało już skóry.

PRZYCZYNY KONFLIKTU

Jakieś 10 lat temu zamorscy kupcy przywieźli do kraju Pol-skah szereg nieznanych tu totემów. Totemy te wyglądały jak niewielkie pudełka wykonane z dziwnego materiału z rzędami wystających płaskich przycisków. Po dotknięciu przycisku działy się niepojęte rzeczy — totem mrugał, zmieniał kolory i wydawał boskie dźwięki w niezrozumiałej mowie.

Różnice między totemami podzieliły żyjące dotychczas, we względnej zgodzie plemiona na zagorzałych przeciwników. Każde plemię chciało udowodnić, że prymat należy się ich totemowi, ponieważ jest on absolutnie najlepszy; doprowadziło to do bratobójczych walk nawet pomiędzy szczepami złączonymi ze sobą dyskiatką przyjaźni.

Tu należałoby Czytelnika wprowadzić w podział etniczny kraju Pol-skah.

Na początku konfliktu do władzy i prymatu doszło plemię Zet-iks (inna nazwa: Sinhk-ler). Na podstawie umów między szczepów żyło ono w przyjaźni z ludem Ti-meks. Do nielicznych w tym czasie należały jeszcze plemiona Athariów i Commo-dhorów, które w późniejszym czasie przejęły na siebie cały ciężar walk. Jedynie plemię niezangażowane w konflikcie (Ai-bi-em) było wystarczająco sil-

ne, aby pozostałe szczepy nawet nie myślały o wchodzeniu na jego tereny myśliwskie.

Konflikt nabrzmiał, gdy agenci wprowadzili do sprzedaży w swoich faktoriach totem przyjęty następnie przez szczep lks-el. Szczep ten był odłamem potężnego plemienia Athariów. Zrazu antagoniści (plemię Commo-dhore, a w szczególności szczep Siks-thy-for) usiłowali zdyskredytować ten totem tolerując go; wnet jednak rozpoczęła się walka na dobre. Athariowie ustąpili początkowo pola nie mając amunicji (zwanej z niezrozumiałych powodów Softh-wahre); za sprawą diabelskich (w pojęciu Commo-dhorów) agentów amunicja ta pojawiła się najpierw w faktoriach, a następnie w słynnej do dziś miejscowości Ghieu-da, w której co tydzień wszystkie plemiona toczyły bratobójczą walkę.

Na dorocznym zjeździe plemion wodzowie postanowili wypowiedzieć wojnę nieuczciwym błędnym twarzom (agentom), a to z powodu zbyt wielkiej liczby skórek bobrowych o rzadkim zielonkawym odcieniu, jakie należało dać za pojedynczy totem. Doszło do spotkania w miejscowości Ghieu-da, gdzie zaczęto (oprócz normalnych walk) wymieniać się totemami i amunicją, co znacząco obniżało zyski agentów.

Mniej więcej w tym samym czasie zaszły głębokie zmiany w strukturach plemiennych. U Athariów oprócz szczepu lks-el wyodrębniły się szczepy lks-e oraz Es-the i Mega-Es-The. Nowy szczep Commo-dhorów (Uan-Tu-Ejt) przegrawszy dyskiatkę pokoju i grę o nieagresji ze szczepem Siks-thy-for, mordował zaciekle swoje własne szczepy Siks-tiin, Plas-for i Vik-tu-enty. Razem natomiast Athariowie i Commo-dhorzy wytypili doszczętnie szczepy obce Em-es-iks, Ak-orn, Than-dy, Bi-bi-si i Spec-thra-vid-heo.

Sytuację pogorszył radykalnie nowy totem plemienia Commo-dhorów, przyjęty przez szczep o nazwie Ammi-ga. Od tej pory jego zwolennicy mordowali wszystkich jak leci, choć nadal nie odważali się zaatakować plemienia Ai-bi-em (pewne aroganckie głosy się jednak podniosły). Plemię Ai-bi-em, co najdziwniejsze, pozostało bierną stroną w całym konflikcie, by w przyszłości okazać się jego przypadkowym zwycięzcą.

Bez pardonów bijatyki pomiędzy Athariami i szczepami Commo-dhorów przybierały tymczasem na sile. Być może mieli na to wpływ wybitni wodzowie tamtych czasów — Wu-Zien-Thara (naczelnik wszystkich szczepów Athariów), Pshas-nys-khi (dowodzący wciąż jeszcze silnym plemieniem Sinhk-ler), czy Em-Pam-puh i Koshier-skhi (obaj ze szczepu Ammi-ga). Do wodzów wybitnych należałoby jeszcze zaliczyć Jot-Mlod-skhi ze szczepu Si-piem (plemię Amst-rhad) oraz Em-Sils-khi czy Fal-khow-skhi ze szczepu Commo-dhorów. Dzięki nim wyjaśniono i obalono wiele mitów o totemach przeciwnika, co niestety przyczyniło się także do pogłębienia waśni i pokrycia bezkresnych preri joystickami męczarni. Binarna krew płynęła strumieniami.

W 1985 roku nastąpił walny zjazd wszystkich plemion, w wyniku którego zaczęto wydawać orędzia i wampumy (BAJ-THEK, KOMPHU-THER, IKX itp.). Parę lat później

ukazały się podobne orędzia przenoszone w poszczególnych plemionach za pomocą wampumów o nazwach: KASHE-TA, DHYSK i MOD-HEM. Niestety, zamiast nawoływać do pokoju ich wydawcy obrzucali się nawzajem wyzwiskami, wyrażając gotowość do pojedynkowania się. Z reguły byli to młodzi wojownicy chcący wykazać się wielką walecznością, aby zdobyć sobie odpowiednio zaszczytne imię w społeczności danego plemienia. Trzeba im przyznać, że posługiwali się niekiedy wprost wyjątkowo nową bronią o nazwie DHE-MO-SY; była to jednak sztuka dla sztuki, ponieważ konkurenci natychmiast oskarżali wybijającego się, że ukradł im wymyślone przez nich chwytły. Takie kłótnie nie przysparzały im zbyt wielu zwolenników.

Konflikt zakończył się w roku 2050, kiedy to przywódca plemienia Ai-bi-em wypowiedział totalną wojnę wszystkim pozostałym szczepom kraju Pol-skah. Po długiej i wyczerpującej bitwie siła totemu Ai-bi-em (a przede wszystkim jego cena) przechyliła szalę na korzyść agresora. Niedobitki Athariów i Commo-dhorów (którzy próbowali nawet zmontować kilka totემów o sile równej totemowi Ai-bi-em, ale nie za bardzo to im wyszło) ukryły się w rezerwach. Przetrwały jedynie plemiona Ammi-ga i MEGA-ES-THE oraz wyznawcy totemów określanych jako 32BITY (prawdopodobnie chodziło o jakiś parametr głównego podzespołu totemu). Mniej więcej od tego czasu skończył się krwawy konflikt w kraju Pol-skah.

Żyjemy w czasach, gdy komputer jest jedną z najbardziej pożądanymi zabawek przez nasze dzieci. Kto nie ma w domu komputera lub nie za bardzo wie, co to jest bit, bywa często dyskredytowany w oczach rówieśników. Zai-

ste trzeba przyznać, że kodeks honorowy amatorów klawiatury jest wyjątkowo drapieżny. Do „najlepszych” i najbardziej godnych podziwu należą ludzie, którzy potrafią złamać zabezpieczenie w grze i wiedzą, jakiego programu użyć, aby skopiować zabezpieczoną dyskiatkę. Na każdym kroku podkreśla się, że najważniejsza jest maszyna, a oprogramowanie do niej to śmieć, który w każdej chwili można kupić gdziekolwiek za śmieszne pieniądze. Autor nowego wirusa zasługuje na co najmniej trzy orle pióra, oczywiście pod warunkiem, że wirus zaatakuję i zniszczy kilkudziesięciu osobom efekty ich pracy (mogą to być np. własne programy użytkownika czy zbiory danych, czego zupełnie nie bierze się pod uwagę). Chęć udowodnienia właścicielowi konkurencyjnego komputera, że jego maszyna to kompletny śmieć, jest tak wielka, że miesza się z błotem nie tylko maszynę, ale również osobę (i niestety osobowość) jej posiadacza. Nikomu jakoś do głowy nie przychodzi, że mogą być ludzie, którzy na taki a nie inny komputer musieli zdecydować się ze względu na kwitnący po wprowadzeniu reform gospodarczych stan portfela, czy też dokładnie przeżyli swoje zastosowania. Smuci mnie fakt, że człowiek rozmawiający ze mną o pewnych problemach związanych z uruchomieniem programu na Spectrum, Atari czy PLUS/4 autentycznie się wstydzi, bo wie, że są „lepsze” komputery. Czy aby na pewno jest się czego wstydzić, panowie? Czy każdy w kraju jeździ mercedesem?

Na szczęście mijają już powoli te czasy, kiedy o wiedzy stanowiła ilość posiadanego oprogramowania (w większości bowiem wypadków sam fakt posiadania programu w żadnym razie nie gwarantuje, że jego właściciel potrafi z niego skorzystać).

Większość użytkowników komputerowych zapomina niestety o fakcie, że ich sprzęt też ma swoje wady (tu pozwolę sobie przytoczyć przykłady z własnego podwórka). Co z tego, że C-64 ma lepszą grafikę, jeśli nie można jej programować w taki sposób jak na Spectrum; co z tego, że Amiga ma kartę emulującą IBM, jeśli koszt samej Amigi i karty znacznie przewyższa cenę standardowego XT z twardym dyskiem. Przenoszenie plików tekstowych i niektórych programów do IBM (dziwnym trafem ostatnio szczegół ten jest jakoś bardzo w cenie) można zrealizować już na ATARI XE, bez konieczności kupowania Atari ST; przykłady można by mnożyć.

Apeluję więc i do Ciebie, Czytelniku: jeśli nawet masz lepszy komputer, to uszanuj bliźniego i nie staraj mu się udowodnić całym sobą, że jest to szmelc, tym bardziej że nie ma jeszcze na świecie komputera, który zakasuje wszystkie inne, tak jak nie ma człowieka bez wad. W większości wypadków bowiem takim ludziom pozornie rozmówca przyznaje rację (chcąc głównie uciąć idiotyczną dyskusję), a w rzeczywistości określa Cię mianem „leszcza”.

A teraz muszę kończyć, bo Wielki Wódz Wu-Zien-Thara w czerwonych barwach wojennych wyzywa mnie na kolejny pojedynek — tym razem na włócznie świetlne. Howgh!

Klaudiusz Dybowski

JUŻ JUTRO TWOJE SPECTRUM LUB TWÓJ TIMEX MOGĄ STAĆ SIĘ SPECTRUM 128k +2, MOGA

➤ PRZERÓBKİ 48KB na 128K +2
➤ NAPRAWY SPECTRUM, TIMEX
➤ NAPRAWY IBM PC XT/AT
➤ NAPRAWY STACJI FDD 3, 3000

P.U.H. » **STAVI** « S.C.
00-227 Warszawa ul. Freta 22/24 m3
inf. listowne: zaadresowana koperta
oraz znaczki za 2000 zł luzem
☎ 31-17-33 10⁰⁰ — 18⁰⁰

KUPNO/SPRZEDAŻ KOMPUTERÓW ➤
KUPNO/SPRZEDAŻ OSPRZĘTU ➤
IBM PC XT/AT, MONITORY i in. ➤
SPECTRUM, TIMEX, AMSTRAD ➤

ŁADOWAĆ NA RAZ WSZYSTKIE "LEVELE" GIER I GENEROWAĆ STEREOFONICZNY DŹWIĘK Z UKŁADU SOUND

NOWE PISMA KOMPUTEROWE:

MOJE **Atari**

dla użytkowników komputerów Atari (w redakcji są do nabycia dyskietki z opublikowanymi programami)

TOP SECRET

dla fanów gier komputerowych!

SZUKAJCIE W KIOSKACH RUCHU!

Wydawca: Spółdzielnia Bajtek, 00-687 Warszawa,
ul. Wspólna 61, tel. 21-12-05

Amiga, C-64 Atari XL, ST

Programy za zaliczeniem
pocztowym
katalogi — gratis
co 5 program — gratis
szybko, tanio oferuje:
Andrzej Zygmunt tel. 588-733
ul. Owocowa 4/9,
40-158 Katowice

B89



poleca naprawy mikrokomputerów
i peryferii

Specjalna oferta:

- rozszerzenia RAM do Amigi 500 — 512 kB (750 tys. zł.)
— 1,8 MB (2,5 mln zł.)
- cartridge do C-64 (150 — 250 tys. zł.)

Dla zamiejscowych naprawy na
poczekaniu.

Ceny orientacyjne.
Gdańsk, ul. Maruszewskiej 6
tel. (058) 48-50-63

B94

DZIAŁ REKLAM:
BAJKA
MOJEGO ATARI
TOP SECRET
codziennie 9—16
21-12-05

P.U. „FORMAT”

01-031 Warszawa, ul. Marchlewskiego 59/73
tel. 38-07-76

oferuje:

Zewnętrzne Stacje Dysków

wszelkich typów (5,25", 3,5", 3")
do komputerów domowych, przenośnych, profesjonal-
nych.

Amiga Atari ST, Amstrad.
Schneider, Toshiba Bondwell, Spectrum,
PS/2, XT, AT i innych.
oraz
Rozszerzenia pamięci do Amigi

B 92

Zakład Usług Elektronicznych „HOMECOMP”

(do niedawna AZUSPHW) poleca
usługi w zakresie serwisu kom-
puterów: Spectrum, C-64, C+4,
Timex, Atari oraz zasilaczy kom-
puterowych.

Warszawa ul. Puławska 102,
tel. 44-87-89
czynny w godz. 11—19, rachunki,
gwarancja.

B95

● ATARAX ●

oprogramowanie,
literatura, instrukcje

ATARI XL/XE
ATARI ST
COMMODORE C-64
COMMODORE 16, 116, +4
AMIGA

sprzedaż wysyłkowa
katalogi gratis, po przystąpieniu zaad-
resowanej koperty zwrotnej +
zaczek.

Adres: ATARAX
05-100 Nowy Dwór Ma-
zowiecki
ul. Chemików 7/15
tel. 75-22-47

B73

ATARI 800 XL, 65 XE, 130 XE

Sprzedaż wysyłkowa
gier i programów użytkowych
na kasetach i dyskietkach.
Również w systemie TURBO
2000
Wszystkie nowości!!!
Instrukcje i literatura.
Dla zainteresowanych rachunki.

ANWIKOL

03-721 Warszawa
ul. Jagiellońska 3/28.

B 81

Atari Turbo 2000 F

Nowy system transmisji danych
z magnetofonem przyspieszony
do 6700 bodów.

Komplet:
— cartridge
— oprogramowanie
— przeróbka magnetofonu
— instrukcja obsługi
— 12 miesięcy gwarancji
Instalacje wykonujemy na po-
czekaniu.

Interfejs do zwykłego magneto-
fonu
Duży wybór oprogramowania w
standardzie TURBO-2000.

Informacja:
Tel. 33-40-91
Korespondencja:
MUEL ul. Człstkowska 30,
01-678 Warszawa.

B 82

Atari XE, XL Spectrum, Timex
niskie ceny programów na taśmie
i dysku
Interfejsy Turbo I "AY" do
Spectrum
Cartridge do Atari — gry i użytki
Informacja — zaadresowana
koperta + znaczek
05-220 Zielonka skr. poczt. 9/2
B80

Programy C-16, C-116, C PLUS 4
C-64 wysyłam pocztą. Katalog
gratis po otrzymaniu koperty
zwrotnej.
Nagrywanie programów Kompute-
rowych
ul. Lenina 104/3
58-304 Wałbrzych

B77

Loader do gier dla Atari współpra-
cujący ze stacją XF 551, umożli-
wiający szybki (Turbo) odczyt z
obu stron dysku jednocześnie.
Cena 50 000 zł.
Zamówienia (realizowane za zali-
czeniem pocztowym) kierować na
adres:

Jacek Szmyd
ul. Wiosenna 15
35-303 Rzeszów

B91

Studio „VIDEO-FAN”

ul. Michałowskiego 17
42-200 Częstochowa, tel. 222238
Commodore 64 — atrakcyjne programy
na kasetach i dyskietkach

B93

ZX SPECTRUM ATARI system turbo, TIMEX FDD 3000,

programy użytkowe, edukacyjne
gry, instrukcje, podręczniki
wysyłka na cały kraj
rachunki
informacje po nadesłaniu
koperty + znaczek.

2"P.K.T.S." Studio Komputerowe
00-103 Warszawa
ul. Królewska 43 m 25

B 79

JOLA

02-117 Warszawa
ul. Racławicka 144 m 112
tel. 56-00-07 godz. 8-15

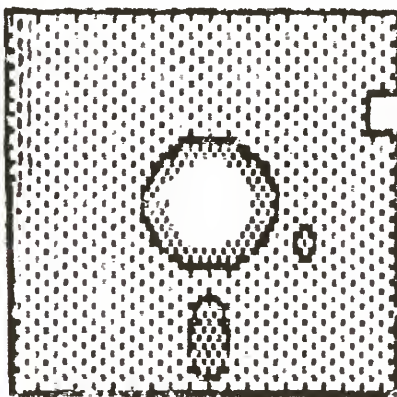
ATARI XE/XL

- Gry i programy użytkowe
- Programy kasetowe i
dyskowe
- Gry w TURBO 2000
- Opisy, instrukcje i
literatura

SZEROKA OFERTA PROGRAMÓW

- TOP DRIVE 1050
- TURBO 2000
- Rozszerzenia pamięci
- Przeróbki magnetofonów
i stacji dysków
- Cartridge BASIC XE, XL,
ACTION
- Bezpłatny katalog B-85
- Najniższe ceny, zniżki
- Zadzwoń lub napisz

	Giełda	Sklep	Pewex	Zachód
	tys. zł		\$	
SINCLAIR				
ZX 81	200	—	—	—
ZX Spectrum 48	800	990	—	—
ZX Spectrum +	900	1100	—	—
ZX Spectrum + 2	1200	—	—	—
Timex 2048	1000	1200	—	—
Stacja FDD 3000	1100	1300	—	—
Stacja FDD 3	800	—	—	—
drukarka GP-50	650	550	—	—
Masterface I	120	—	—	—
AY 3-8910	170	—	—	—
COMMODORE				
C 64	1900	1950	207	—
C 128	2500	2700	—	—
C 128 D	4000	4200	599	—
Amiga 500	5000	5800	795	—
Amiga 2000	12000	—	—	—
Amiga 3000	—	—	—	—
magnetofon	350	400	30	—
stacja 1541	1800	1900	—	—
stacja 1571	2200	—	—	—
stacja Oceanic	1900	—	—	—
drukarka LC 10C	2100	—	299	—
Final II	100	120	—	—
Final III	230	250	—	—
Action Replay	350	—	—	—
ATARI				
Atari 800 XL	1200	1250	—	—
Atari 65 XE	1500	1600	159	—
Atari 130 XE	1900	2200	239	—
Atari 520 STFM	—	4300	499	—
Atari 1040 STFM	—	—	899	—
magnetofon	300	350	51	—
stacja CA 2001	2600	—	249	—
monitor SM124	2200	—	—	—
monitor SM224	—	—	480	—
Turbo 2000	100	100	—	—
Centronics	200	220	44	—
AMSTRAD				
Amstrad 464	2000	2200	—	—
Amstrad 6128	2700	2900	—	—
Amstrad PCW 8256	—	—	—	—
Amstrad PCW 8512	3600	—	—	—
IBM				
IBM PC XT stand.	8000	9500	499	—
IBM PC AT stand.	11000	—	789	—
IBM PS/2	—	16000	—	—
HD 20 MB z kontrol.	3100	—	260	—
napęd 5.25"	600	700	—	—
monitor amber	—	—	219	—
klawiatura	440	—	—	—
INNE				
dyskiety 5.25"	4.5	4.5-8	0.5-3	—
dyskiety 3.5"	15	15-20	1-3	—
dyskiety 3"	29	35-45	—	—
kasety C-60	10	—	1	—
monitor	700	750	—	—
joystick	60-80	60-100	7-11	—



INDYWIDUALNY BANK DANYCH

Kamil Ratajczak lat 13, posiada Commodore 64 oraz magnetofon. Oprogramowanie: około 300 gier i 60 programów użytkowych. Pragnie nawiązać kontakt w celu wymiany doświadczeń i oprogramowania. Adres: ul. Buczka 4/13, 14 100 Ostróda

Kacper Dolatowski, lat 15. Posiada Amigę 500.

Oprogramowanie: ponad 200 gier oraz programów użytkowych. Proponuje wymianę oprogramowania oraz doświadczeń. Udzieli również wskazówek dotyczących rozszerzenia pamięci Amigi do 1 MB, odpowie na każdy list.

Adres: ul. Tarnobrzaska 1/7, 53-404 Wrocław.

Miodek Grzegorz lat 15, posiada Commodore 64 (nowa wersja) magnetofon DR 1535, około 300 gier. Zainteresowania: muzyka komputerowa, informatyka, elektronika.

Proponuje wszystkim wymianę oprogramowania.

Adres: ul. K. Wielkiego 2A/8, 47-220 Kędzierzyn-Koźle.

Michel Niski lat 17. Posiada komputer Apple IIe, monitor, 2 stacje dysków, drukarkę Dot Matrix Printer, modem Hayes Smartcom I, Soft Card System, 80 Column Text Card Manual oraz około 100 dysków z programami.

Nawiąże kontakt z posiadaczami Apple w celu wymiany oprogramowania, literatury i doświadczeń.

Adres: ul. Mickiewicza 13 b, 83-200 Starogard Gdański.

Michał Kwolek posiada ZX Spectrum i około 100 programów gier. Pragnie nawiązać kontakt w celu wymiany programów. Adres: ul. Okružni 920, Orlova Lutyne 735 14, CSFR.

Mariusz Kotecki, lat 13. Posiada Commodore +4 z magnetofonem. Oprogramowanie: programy muzyczne, graficzne, użytkowe i około 100 gier. Nawiąże korespondencję w celu wymiany oprogramowania i doświadczeń. Adres: 11-021 Jedzbark, Mokiny 13, woj. Olsztyńskie.

Marcin Wirkus lat 15. Posiada komputer Laser 500 wraz z magnetofonem K 40 Elektronik. Pragnie nawiązać kontakt z użytkownikami tego komputera. Adres: 76-231 Damnica, ul. Wiśniowa 2, woj. Śląskie.

Piotr Ornass — uczeń 41 kl. LO. Posiada Amigę 500. Prosi o kontakt w celu wymiany oprogramowania, literatury, i doświadczeń. Adres: 89-410 Więcbork, ul. Mickiewicza 23

Kastelnik Mieszko lat 15. Posiada Commodore 64, magnetofon Datasette 1530, Cartridge Final 3 oraz 1000 programów. Proponuje wymianę literatury i oprogramowania. Adres: 32-300 Olkusz ul. Armii Czerwonej 16.

Leszek Jastrzębski lat 14, posiada Commodore Amigę. Pragnie nawiązać kontakt w celu wymiany literatury i doświadczeń. Adres: 41-800 Zabrze ul. Warszawska 42

Jacek Kuchta lat 13 posiada Atari 800 XL system 2000 i 2001, z magnetofonem XC, 12. Pragnie nawiązać kontakt w celu wymiany gier. Warszawa 02-777, ul. Puszczczyka 4/35.

joy

**Wysyłka natychmiastowa
za zaliczeniem pocztowym**
Joysticki do Atari, Commodore, Spectrum, Amstrad.

Precyzyjny mechanizm, specjalne styki.

Kable z wtyczką, przedłużacze do joysticków.

Interface do Spectrum. 6 miesięcy gwarancja.

Elektromechanika

ul. Cegielniana 17
32-410 Dobczyce

B97



Cieślakowski i s-ka
PRZEDSIĘBIORSTWO HANDLOWO-USŁUGOWE
00-446 WARSZAWA UL. FABRYCZNA 2/103
☎ 29-89-31

OFERUJEMY PERYFERIA

AMSTRAD

ATARI ST

AMIGA

Stacje dysków 5.25"

Rozszerzenia pamięci

Modulatory TV

Kontroler stacji dysków CPC 464

Karta EPROM-ów CPC

RS 232 CPC

RS-CENTRONICS PCW

INTERFEJS joystick'a do PCW

8-bitowy CENTRONICS CPC

VIDEO DIGITIZER ST

PROGRAMATOR EPROM-ów

Drogi Bajtku!

Na listy Czytelników odpowiadają autorzy „Bajtka”

Do napisania tego listu skłonił mnie artykuł autorstwa pani Anety Bryske, pt. „TOS inaczej” zamieszczony w *Bajtku* nr 9—10/90. Skądinąd wiadomo mi, że Aneta Bryske to pseudonim artystyczny red. Macieja Pietrasia. Jako autor systemu TOS VA.4 chciałbym odpowiedzieć na zarzuty do systemu zawarte w tym artykule.

1. **Mało przyjazny w obsłudze.** Duże grono znanych mi użytkowników nowego systemu nie wnosi żadnych zastrzeżeń co do sposobu obsługi i ogólnego wyglądu TOS'u VA.4. Przejrzyste i efektowne MENU pozwala w prosty sposób skorzystać z każdej opcji systemu. Po naciśnięciu klawisza N możemy wrócić do BASIC'a. Wtedy wszystkie jego instrukcje pozostają takie same jak w systemie TOS VA.2.
2. **Częste skłonności do zwieszania się.** Podczas ponad półrocznej pracy z nowym systemem nie zdarzyło się znanym mi użytkownikom, aby system zawiesił się. Spadek napięcia może spowodować zawieszenie się systemu TOS VA.2 tak, jak i TOS VA.4.
3. **Możliwość uszkodzenia przechowywanych danych.** Prawdopodobieństwo uszkodzenia danych na dyskietce jest takie samo, jak w systemie TOS VA.2. Natomiast możliwość uszkodzenia dyskietki jest mniejsza. Dzieje się tak dlatego, że TOS VA.2 doczytuje za każdym razem moduły systemu z dysku, natomiast TOS VA.4 jest całkowicie ulokowany w pamięci RAM stacji dysków.
4. **Chęć powrotu do prostych TOS'owych poleceń.** Moim zdaniem jest to bezpodstawne stwierdzenie. TOS VA.4 został pomyślany tak, aby mógł się nim posługiwać każdy użytkownik, nawet ten, który niewiele zna się na instrukcjach rozszerzonego BASIC'a. Formatowanie dyskietki, kasowanie i zmiana nazwy plików, i inne udogodnienia nie wymagają powrotu do BASIC'a, i są dostępne z poziomu programów użytkowych znajdujących się w pamięci RAM stacji dysków.
5. **Programy stale dostępne dla użytkownika.** Zaskakującym jest fakt, że wśród programów użytkowych stale dostępnych dla użytkownika, znalazł się program o nazwie Disk Editor. Istniała wersja systemu z tym programem, ale nie była ona ogólnie udostępniona i autor artykułu również jej nie posiadał.
6. **Brak pracy z podkatalogami.** Jest to jedyny prawdziwy zarzut dla systemu TOS VA.4. W artykule czytamy jednak: „mała pojemność dyskietki sprawiła, że hierarchiczny system zbiorów jest raczej ciekawostką i nie może być w pełni wykorzystany”. Trudno nazwać 164 kB wolnej pamięci na jednej stronie pamięcią wystarczającą na zakładanie podkatalogów. Jeżeli jednak chodzi o pracę z podkatalogami, to jest to w systemie TOS VA.4 możliwe. Jedyny problem jaki może się pojawić to taki, że program START pokazuje zawsze pliki z podkatalogu nr. 0. Napisanie programu START w taki sposób, aby dawał nam możliwość poruszania się po drzewiastej strukturze katalogu nie jest trudne i jest do wykonania.

Aktualnie powstała nowa, zmodyfikowana wersja systemu, wzbogacona o kolejne programy użytkowe i możliwość korzystania przez użytkownika z poziomu BASIC'a z około 48 kB pamięci RAM stacji dysków.

Podsumowując, nowy system jest bardzo przydatnym udogodnieniem nie tylko dla graczy, lecz dla każdego posiadacza RDD 3000.

Radosław Cymer

Od początku roku jestem posiadaczem AMSTRADA 6128 z drukarką D-100-M. Aby sprzęt był coś wart, kupiłem kilka programów w Agencji Komputerowej „BETA B” z Sosnowca. Zdaję sobie sprawę z nielegal-

ności ich kopii, ale skąd brać oprogramowanie do „nie-IBM-a” za złotówki? Ale do rzeczy.

Wiadomo, że rozpoczyna się kupowanie oprogramowania od m.in. edytora tekstu. Ponieważ opisy w „Bajtku” czy „Komputerze” są skąpe, a firmy dysponują w zasadzie jedynie tytułami, kupiłem „Mini Office II”. Po bardzo długim „wgryzaniu się” (pełna empiria), przejrzałem go za pomocą „Discology”.

To, co zobaczyłem skłania mnie do obaw, czy nie jest to WIRUS?! Myślałem, że wirusy dotyczą komputerów IBM, skąd wziąć program antywirusowy dla komputerów 8-bitowych?

Przy okazji, inna sprawa — nie dojdziemy do niczego w komputeryzacji społeczeństwa, bo między informatykami a szarym użytkownikiem jest zupełna dziura. Nawet za dziesięć lat nie będę sprawnie wykorzystywał oprogramowania systemowego posługując się jedynie „helpem”. To jeden z przykładów. Ponieważ nie opłaca się tworzenie oprogramowania, to może chociaż opisy po polsku.

Wit Dobrowolski

Współczuję Panu z powodu używania D-100-M i całkowicie podzielam pogląd o nielegalności oprogramowania. Można to rozwiązać tylko przez zmianę prawa autorskiego, gdyż (na razie) piratów nie można (poza kilkoma przypadkami) skutecznie zmusić do zaprzestania procederu.

Co do podejrzeń, to osobiście nie wierzę w wirusa w „Mini Office”. Nikt (absolutnie nikt z przesłuchanych przeze mnie osób) nic nie słyszał o wirusach na Amstradzie. Zachodnie magazyny także nie ostrzegają przed wirusami, co na pewno robiłyby, gdyby takowe istniały. Problem wirusów nie dotyczy tylko IBM PC („amiganci” także mają z nimi duże kłopoty, podobnie Atari ST, a nawet Commodore 64). Systemy operacyjne Amstrada są tak prymitywne, że nikomu nie chciało się pisać wirusów (dużo pracy i wątpliwy efekt), poza tym Amstrady są zbyt mało popularne, by zyskać sławę tworząc wirusa. Jeśli chodzi o programy antywirusowe, to po prostu ich nie ma. Jeśli jednak uważa Pan, że należy dokładniej zbadać podejrzany fragment, prosimy o przysłanie dyskietki (gwarantujemy zwrot).

Ma Pan rację, „helpy” nie wystarczą. Problem jest w tym, że nie ma kto napisać tych opisów. Pisanie i wydawanie opisów oprogramowania na komputery 8-bitowe jest nieopłacalne, poza tym za pisanie bierze się głównie „Bielecki i S-ka”, produkując bezsensowny bełkot. Może wyjdzie stworzenie klubu użytkowników i pisanie opisów przez osoby znające dany program.

M.SZ.

MICROMAN

oferuje na miejscu lub wysyłkowo:

1. Sprzęt komputerowy i akcesoria.
2. Programy i literaturę do komputerów: Atari XL/XE/XT, Commodore 16/116/+4/64/125/ Amiga, Spectrum, /Timex.
3. Oprogramowanie na cartrig'e.
4. Przystawki „UNIVERSAL TURBO” do magnetofonów firmowych Atari umożliwiające zapis i odczyt programów w systemie Blizzard oraz Turbo 2000
5. Naprawy zasilaczy, magnetofonów, klawiatur w komputerach Atari Commodore Spectrum.

Informacje: (na miejscu lub koperta zwrotna), zamówienia:

- 40-181 Katowice ul. Osikowa 66 tel. 585-106
- 44-200 Rybnik ul. Wiejska 19 tel. 233-56

Firmowe punkty sprzedaży:

- Katowice ul. Plebiscytowa 31
- Rybnik D.H. „Hermes” I piętro
- Bielsko-Biała Plac Wolności 9

B74

B88

KUPIĘ • SPRZEDAM ZAMIENIĘ

Każdy, kto przyśle do nas dwa, wycięte z kolejnych numerów *Bajtka* kupony (odbitek nie będziemy honorować), może zamieścić krótkie ogłoszenie. Maksymalna długość ogłoszenia — piętnaście słów razem z adresem, drobne odchylenia do zaakceptowania. Ogłoszenie może dotyczyć sprzedaży, kupna lub zamiany komputera i akcesoriów — wszelkiego typu urządzeń zewnętrznych, programów i literatury, używanych i nowych, pod warunkiem, że oferta dotyczyć będzie pojedynczych sztuk. Ogłoszenia drukować będziemy kolejno w miarę ich napływania. Zastrzegamy sobie prawo niewydrukowania ogłoszenia zbyt długiego, a także w razie podejrzeń o próbę sprzedaży hurtowych ilości towaru lub kradzionego oprogramowania, oraz prawo do zmiany zasad akceptowania ogłoszeń. Piszcie na nasz adres, z dopiskiem na kopercie — **Kupię-Sprzedam-Zamienię.**

1. Kupię każdą grę na komputer ZX81. Michał Jurkowski, ul. Miklaszewskiego 6/2, Warszawa, tel. 643-16-93.
2. Kupię ATARI XEP 80. Rafał Wąsiewski, 06-100 Pułtusk, ul. Wiodok 12/20.
3. Kupię „KOMPUTER” 4/90, FDD 3000+TOS V1.2. ZX Spectrum — wymienię programy. Rafał Szamocki, 75-454 Koszalin 12, skr. poczt. 12.
4. Sprzedam Commodore 64C, stację dysków 1541-II, Joystick. Kielce, tel. 31-02-53.
5. Programy na CPC 6128, wymienię, kupię, odstąpię. M. Szmolke, ul. Kuźnia 10, 30-444 Kraków.
6. Kupię literaturę o Atari 65XE. Radek Piesiewicz, ul. Powstańców 7, 09-402 Płock.
7. Sprzedam książki o ZX Spectrum. Karpeta Krzysztof, 36-020 Tyczyn, ul. Słowackiego 3.
8. Wymienię programy — ZX Spectrum 48. Marek Kałafarski, skr. poczt. 3, 33-106 Tarnów 8.
9. Kempston (RESET) sprzedam, 60 tys. zł, Marcin Śmietana, ul. Robotnicza 2/86, 39-200 Dębica.
10. Sprzedam lub zamienię ATARI 800 XL, 1010 z TURBO 2000 na C-64 z magnetofonem. Michał Bikowski, Ostrów Maz., tel. 40-09.
11. Kupię program do projektowania obwodów drukowanych. Artur Fryczek, Cikowice 137, 32-700 Bochnia.
12. Sprzedam ATARI 800 XL, stację LDW SUPER 2000, 91 dyskietek z pudełkiem. Jarosław Statkiewicz, Baśniowa 28, 07-412 Ostrołęka.
13. Kupię oprogramowanie do New Brain oraz literaturę fachową, Warszawa, tel. 39-58-53
14. Sprzedam Atari 65XE, stację CA-2001, magnetofon XCA12, dyski + programy, joystick Michał Mielech, Białystok 21, skrytka 49.
15. Kupię gry PLATOON i BRUCE LEE na Atari 800XL ze stacją dysków. Piotr Bachar, Gdańsk Wrzeszcz, ul. Waryńskiego 12/5.
16. Kupię tanio stację dysków FDD 3000, Jacek Nakielski, Gdańsk, ul. Wyspiańskiego 5a.
17. Wymienię pocztą gry na ZX Spectrum. Maciej Kozłowski, 04-088 Warszawa, ul. Majdańska 3/89.
18. Sprzedam (całość): Commodore 64C, 1530 Datasette Unit, 2 Quick Shot-y I, Final II + 20 kaset, Mariusz Rewkowski, 71-664 Szczecin, Jarogniewa 30/5, tel. 522-777.
19. Kupię Atari 520 STFM (dwustronna stacja dysków). Oferty z ceną: Andrzej Stechnij, Stargard Szczeciński, ul. Płatnerzy 3c/10, tel. (092)77-34-91.
20. Czarną motorynkę (stan idealny), zamienię na komputer lub sprzedam. Darek Lange, 05-200 Wołomin, ul. Armii Ludowej 80, tel. 76-22-74.
21. Kupię nowy Commodore 64 i gry. Obuchowski Tomasz, 82-300 Elbląg, ul. Kościuszki 31/36.
22. Kupię komputer Atari 65XE z magnetofonem XC 12. Grzegorz Bilewski, 18-400 Łomża, ul. Kazańska 7/16.
23. Sprzedam Schneider 6128, kolorowy monitor, literatura, dyskietki systemowe — 4.5 mln zł. Eryk Szuba, 04-175 Warszawa, ul. Ostrobramska 78/43.
24. Sprzedam drukarkę do kalkulatora T159 z zapasem papieru. Marzena Popis, 04-526 Warszawa, ul. Króla Maciusia 9/9.
25. Pilnie kupię numery *Bajtka* 3—4/86, 11/89. Bartosz Rybiński, 04-410 Warszawa, ul. Szatkowników 10.
26. Kupię fabrycznie oryginalny interfejs joysticka do Spectrum +. Tomasz Zwierzycki, 67-100 Nowa Sól, Os. XXX-lecia PRL 5/1m.
27. Sprzedam Commodore C-128, stację 1570, oprogramowanie, literaturę, cena 4 mln. zł. Sopot, tel. 51-03-55, codziennie 9—15 prócz sobót i niedziel.
28. Kupię FDD 3000. Sprzedam, kupię, wymienię programy na AY. Daniel Drozd, 81-611 Gdynia, ul. Wielkokącka 12/33.
29. Sprzedam Spectrum-48 w bardzo dobrym stanie, pełny osprzęt. Paweł Laskowski, 09-100 Płońsk, ul. Płocka 19.
30. Dam wszystko za Ultima IV bez błędów. Wymiana dysków na Atari. Jarosław Warylewski, Gdynia 4, skr. 64.
31. Odsprzedam lub wymienię oprogramowanie na Atari XE/XL w systemach AST oraz Turbo 2000F. Wojciech Todryk, 16-100 Sokółka, Os. Centrum 3/19.
32. Kupię uszkodzone Spectrum lub samą płytkę. Mariola Wandycz, 59-300 Lubin, Plac Wolności 14/3.
33. Sprzedam Atari 65XE, LDW2000, XC12. Kupię IBM, Andrzej Stosur, Tarnów, ul. Limanowskiego 16/4, tel. 220-524.
34. Sprzedam literaturę do ZX Spectrum (Basic, Logo, Asembler). Gdańsk, ul. Pomorska 22b/25, tel. 57-94-24.
35. Sprzedam komputer Timex 2048, magnetofon Timex, joystick. Tomasz Wróblewski, 41-200 Sosnowiec, ul. Mielczarskiego 8/21, tel. 66-94-15.
36. PC XT 10 MHz tanio sprzedam. Marcin Kalisz, Warszawa, tel. 25-44-81 wew. 462, poniedziałek i środa, 8—15.30.
37. Kupię oprogramowanie do projektowania w zakresie inżynierii sanitarnej do PC/AT. Andrzej Traczyk, 54-432 Wrocław, ul. Strzegomska 280/2, tel. 577-193.
38. Kupię nową tabliczkę graficzną „KoalaPad” do Atari XE. Maciej Grzeszczuk, 02-381 Warszawa, ul. Trzech Budrysów 33/23.
39. Wymienię oprogramowanie do Atari 520/1040 ST, Sharp MZ-700/800. Ryszard Treła, 42-500 Będzin, skrytka 15.
40. Poszukuję książki „Atari BASIC” W. Miguta. Seweryn Bóldak, 15-869 Białystok, ul. Białostockie 14/33.
41. Bardzo tanio sprzedam drukarkę D-100. Pilne. Sławomir Kierzkowski, 96-300 Żyrardów, ul. Rodzinna 1/30.
42. Sprzedam Atari 65 XE, drukarkę Atari 1029, stację LDW 2000. Hubert Radej, 04-489 Warszawa, ul. Gembarzewskiego 2.
43. Sprzedam ATARI 65 XE, XC 12 z Turbo, AST cartridge, joystick, monitor. Artur Nózka, 20-353 Lublin, ul. Bukowa 21/4.
44. 4-letni magnetofon Atari XC 12 z Turbo 2000, stan dobry, sprzedam. J. Gomułka, 45-061 Opole, ul. Dekabrystów 12/3.
45. Tanio sprzedam Timex 2048 z magnetofonem. Piotr Sztlauch, 24-100 Puław, ul. Próchniaka 4/49.
46. Sprzedam komputer Amstrad CPC-464 z zielonym monitorem + instrukcja. Andrzej Weber, Warszawa, tel. 31-77-88 w godz. 21.00—9.00.
47. Kupię dwukieszeniową stację 3-calową FDD-3000 oraz Timex 2048, najchętniej nowe. Mariusz Dzięgiel, 66-400 Gorzów Wielkopolski, ul. Armii Czerwonej 64/10.
48. Sprzedam używany magnetofon do Commodore. Oferty z ceną: Tęcza Hubert, 86-304 Grudziądz, JW 5794 „0”.

KOMPUTER NATYCHMIAST
KUPISZ-SPRZEDASZ
MAXSOFT
659-44-17 Warszawa

B74

B88

kupię-sprzedam
zamienię
2

SV 119 – 64 900	SV 123 – 119 900	SV 128 – 319 900	SV 210 – 219 000
SV 120 – 74 900	SV 125 – 239 900	SV 130 – 339 000	SV 500 – 119 000
SV 122 – 89 900	SV 126 – 169 900	SV 201 – 249 000	SV 510 – 119 000
SV 124 – 109 900	SV 127 – 249 900	SV 202 – 229 000	

sprzedaż hurtowa
tel. 23-98-53

SV 201 + SV 210 – 439 000
SV 202 + SV 210 – 399 000

udziela informacji
tel. 662-35-16

CENY DETALICZNE AKTUALNE DO 31 STYCZNIA 1991 r.

JAK ZAMAWIAĆ?

NALEŻY PRZESŁAĆ PIENIĄDZE PRZEKAZEM TELEGRAFICZNYM (DUŻY ZIELONY BLANKIET) na adres:

„TAL” Sp. z o.o.
ul. Mikowa 45
02-411 WARSZAWA

UWAGA! W miejscu na KORESPONDENCJĘ prosimy PODAĆ PO RAZ DRUGI: adres domowy, kod i symbol zamówionej pozycji.

CZAS REALIZACJI ZAMÓWIENIA 3-6 dni.

W PRZYPADKU BRAKU ADRESU ZWROTNEGO, NIE REALIZUJEMY ZAMÓWIENIA.

DO KAŻDEGO JOYSTICKA DOŁĄCZAMY KUPON KONKURSOWY

PROMOCYJNA SPRZEDAŻ KOMPUTERA AMIGA 500 i COMMODORE 64 — wersja angielska!!!

DDD — Dostawa Do Domu!!!
szczegóły tel.: 662-35-16

**365 dni
GWARANCJI**

Adres firmy: „TAL” Sp. z o.o., Mikowa 45,
02-411 Warszawa Włochy

tel. 23-98-53 sprzedaż hurtowa
fax: 659-12-35
tel. 662-35-16 udziela informacji

szukaj znaku



tam znajdziesz joysticki



**SV 119
Junior**
2 Fire
6 Blaszanych styków
Prosty mechanizm



**SV 120
Junior-Stick**
2 Fire
6 Blaszanych styków
Uchwyt pistoletowy



**SV 122
Quickjoy II**
2 Fire
6 Blaszanych styków
AutoFire
Drażek lotniczy



**SV 124
Turbo**
6 Mikrostryków
AutoFire
Drażek lotniczy



**SV 123
Supercharger**
2 Fire
6 Mikrostryków
Ergonomiczna budowa
Precyzyjny mechanizm



**SV 126
Jet Fighter**
2 Fire
6 Mikrostryków
AutoFire
ACS-Regulator
szybkości AUTO
Obsługa pod kciuk
Drażek lotniczy



**SV 125
Superboard**
6 Fire
10 Mikrostryków
AutoFire
Cyfrowy wyświetlacz
czasu
Sygnał dźwiękowy
Przełącznik dla
leworęcznych
Drażek lotniczy



**SV 130
IR Infrared**
1 Fire
5 Mikrostryków
Podczerwień
Daleki zasięg
Odbiornik



**SV 128
Megaboard**
4 Fire
10 Mikrostryków
AutoFire
6 cyfrowy stoper
ATM — Anti Tilt Mechanism
Fire Pad

**SV 140
Enterprise**

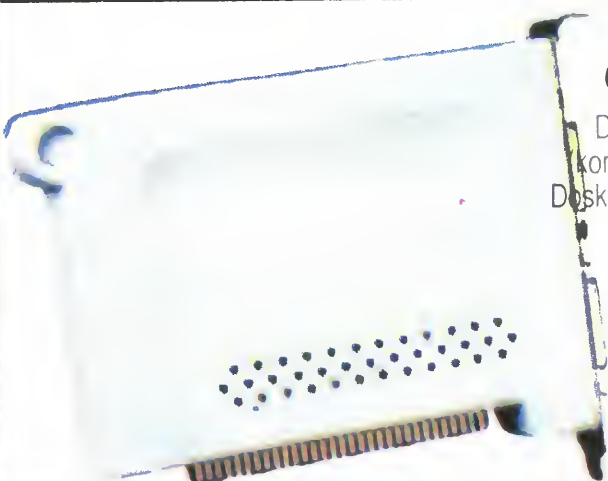
2 Fire
6 Mikrostryków
AutoFire
ACS — Regulator
szybkości AUTO
Drażek lotniczy
„kierownica”
Kabel 4 m



**SV 201
Quickjoy M 5**
Do IBM XT/AT
(kompatybilnych)
Współpracuje z Game-Card
lub I/O Card
2 Fire
2 AutoFire
6 Mikrostryków
Wybór AUTO
PSC — Regulator XY
Sygnalizacja Świecna
Fire
ASC — Regulator szybkości
AUTO



**SV 202
M 6 analog**
Analogowy
DO IBM XT/AT
(kompatybilnych)
Współpracuje z Game-Card
lub I/O Card
2 Fire



**SV 210
Game Card**
Do IBM XT/AT
(kompatybilnych)
Doskonale pracuje
z M 5 i M 6



**SV 127
Top Star**
2 Fire
6 Mikrostryków
AutoFire
Przezroczysta obudowa
SAS — Shock Absorbing
System
Platynowane części



**SV 500
Van 3**
Pudełko na dyskietki
80 sztuk 3 1/2"
Zamknięcie na klucz



**SV 510
Van 5**
Pudełko na dyskietki
80 sztuk 5 1/4"
Zamknięcie na klucz

Quickjoy

TAL — najtaniej w Polsce!



PACKET

RADIO

Miłośnicy transmisji danych komputerowych przez łącza telefoniczne nie mają w Polsce łatwej egzystencji. Modem umożliwiający taką transmisję jest już drogim urządzeniem, dodatkowo należy płacić abonament za użytkowanie modemu i każde połączenie. Fatalna jakość łącz powoduje częste błędy w transmisji, co bardzo ją wydłuża, a niekiedy wręcz uniemożliwia jej realizację.

Długie transmisje, szczególnie z odległymi miastami lub z zagranicą, są więc dość kosztowne. W jaki sposób temu zaradzić, co zrobić, jeśli nie ma się telefonu i żadnych perspektyw na jego otrzymanie w najbliższej przyszłości? Odpowiedź jest prosta — PACKET RADIO.

Packet Radio jest nowym rodzajem przesyłania danych, w którym transmisja odbywa się nie przez sieć telefoniczną, lecz przez radio. Łączność może odbywać się na falach ultrakrótkich lub krótkich w pasmach częstotliwości przydzielonych dla krótkofalowców. Pasma UKF zwykle jest wykorzystywane do łączności lokalnych (do ok. 60 km), a pasmo KF na znacznie większe odległości.

Zasięg w paśmie UKF można poszerzyć wykorzystując stacje przekaznikowe. Są to automatyczne urządzenia, których zadaniem jest odebranie docierającego do nich sygnału, wzmocnienie go i ponowne wyemitowanie. Dodatkowo każda pracująca stacja Packet Radio może służyć jako przekaznikowa dla innej. Operator może przestać dane nawet przez kilka takich stacji.

W wyposażenie niezbędne do pracy składa się z radiostacji nadawczo-odbiorczej (transceivera), specjalnego modemu i komputera. Modem — to zazwyczaj oddzielny mikrokomputer współpracujący z komputerem głównym przez łącze RS 232. Realizuje on wszystkie niezbędne funkcje do przeprowadzenia łączności. Od komputera i programu z nim współpracującego zwykle wymaga się jedynie, aby dane przesyłane z modemu wyświetlał na ekranie, wysyłał zaś to, co napiszemy na klawiaturze.

Ponieważ jedna częstotliwość pasma radiowego może być wykorzystywana przez wielu użytkowników, informacje przesyłane są w specjalny sposób. Blok danych jest dzielony na części o maksymalnej długości równej 256 bajtów. Do każdej takiej części modem automatycznie dodaje znaki wywoławcze adresata i nadawcy, 16-bitową sumę kontrolną CRC, znaki synchronizacji i inne. Tak uformowany blok danych nazywa się pakietem lub paczką i może być wysyłany. Nadawca odbierając pakiet sprawdza jego prawidłowość. W razie prawidłowego odebrania zostaje on potwierdzony przez odbiorcę poprzez nadanie specjalnego pakietu kwitującego do stacji nadawczej.

Po prawidłowym odebraniu pakietu stacja nadawcza może rozpocząć nadawanie następnego pakietu. Pakiety są numerowane, co umożliwia ich właściwe uporządkowanie po stronie odbiorczej. W razie stwierdzenia nieprawidłowości odbiorca żąda powtórzenia tego fragmentu (również za pomocą specjalnego pakietu). Oczywiście kwitowanie każdej porcji danych spowalnia transmisję i dlatego dopuszcza się jednokrotne kwitowanie do 7 przesłanych pakietów. Umożliwia to lepsze wykorzystanie kanału radiowego, szczególnie przy braku zakłóceń.

Oczywiście w skrajnych przypadkach powtarzanie jednego pakietu mogłoby trwać w nieskończoność, ogranicza się to przez określenie maksymalnej liczby prób transmisji. Po wyczerpaniu dopuszczalnej liczby powtórzeń modem przerywa połączenie.

Adresowanie przesyłanych danych pozwala modemowi stwierdzić, czy aktualnie odebrany pakiet jest przeznaczony do niego, czy też nie. Modem nie dekoduje danych nie przeznaczonych dla niego, zapewniając selektywny odbiór. Oczywiście możliwe jest wymusze-

nie odbioru wszystkich przychodzących pakietów, jak też nadanie takich, które odbiorą wszyscy.

Oczywiście zwykły użytkownik systemu nie musi znać tych informacji, są one bowiem automatycznie wykonywane przez modem. Do użytkownika należy jedynie wydawanie prostych poleceń, jak na przykład połączenie się z konkretnym użytkownikiem, ustalenie parametrów transmisji, wysłanie depeszy.

Możliwość nadawania depesz do wszystkich użytkowników jest bardzo cenna, umożliwia bowiem tworzenie i pracę skrzynek kontaktowych (ang. mailbox). Skrzynkę kontaktową tworzy specjalnie przeznaczony do tego celu komputer z odpowiednim programem oraz oczywiście modem i radiostacja. W stanie oczekiwania na połączenie komputer wysyła do wszystkich użytkowników co pewien czas tekst zachęcający do łączności z nią. Jeśli połączymy się, to na ekranie komputera zobaczymy wizytówkę skrzynki oraz jej menu. Zazwyczaj można wtedy przejrzeć wiadomości ogólne, tzn. dla wszystkich użytkowników, zostawić depeszę przeznaczoną dla konkretnego odbiorcy, operatora skrzynki lub też wszystkich. Oczywiście jeśli ktoś zostawił depeszę dla Ciebie, to zostaniesz o tym specjalnie poinformowany.

Skrzynki kontaktowe pracują zwykle całą dobę i są bardzo użyteczne, jeśli chcemy przekazać wiadomość komuś, kto aktualnie nie pracuje lub dla szerszej grupy odbiorców. Inteligencja skrzynki zależy oczywiście od programu realizowanego przez komputer. Często zatem oferują one o wiele więcej możliwości niż wymienione.

Transmisje Packet-Radio są bardzo popularne w USA, krajach Europy Zachodniej, a także na Węgrzech i w Jugosławii. Funkcjonuje tam również wiele skrzynek kontaktowych. W Polsce praca w tym systemie dopiero się rozwija. Przykładowo w Warszawie czynne są już dwie skrzynki kontaktowe pracujące w paśmie UKF.

W jaki sposób rozpocząć pracę? Nie jest to droga łatwa! Aby móc używać radiostacji amatorskiej, należy uprzednio uzyskać odpowiednie zezwolenie. W praktyce polega to na zdaniu egzaminu przed komisją PIR i uzyskaniu licencji krótkofalowca. Jeśli już ją uzyskamy, wystarczy zakupić radiostację i specjalny modem, podłączyć całość do komputera (IBM lub Commodore), uruchomić odpowiedni program komunikacyjny, jak na przykład LAN-LINK, aby rozkoszować się do woli połączeniami.

Cena wyposażenia niezbędnego do pracy (modem i radiostacja) jest wprawdzie około dwukrotnie wyższa od zwykłego modemu telefonicznego, ale za połączenia nie płaci się ani grosza, nawet z USA!

Robert Magdziak
SP5 UGO